

JOSEF STANISLAW FALCÓN MAGALHÃES

SISTEMA VARIANTE PARA SELEÇÃO DE MÁQUINAS-FERRAMENTA

**CURITIBA
2008**

JOSEF STANISLAW FALCÓN MAGALHÃES

SISTEMA VARIANTE PARA SELEÇÃO DE MÁQUINAS-FERRAMENTA

Dissertação apresentada como requisito parcial
para obter o título de Mestre em Engenharia
Mecânica do curso de Mestrado em Engenharia
Mecânica da Universidade Federal do Paraná,
na área de concentração de Manufatura.

Orientador: Prof. Dr. Dalberto Dias da Costa

CURITIBA
2008

Magalhães, Josef Stanislaw Falcón
Sistema variante para seleção de máquinas-ferramenta / Josef
Stanislaw Falcón. – Curitiba, 2008.
72 f. :30 il.,4 tabs,

Orientador: Prof. Dr. Dalberto Dias da Costa
Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de
Tecnologia, Curso de Pós-Graduação em Engenharia Mecânica.
Inclui Bibliografia.

Máquinas-ferramenta - Seleção. 2. Máquinas-ferramenta – Base
de dados. 3. SQL (Linguagem de programação de computador).
I. Costa, Dalberto Dias da. II. Título. III Universidade Federal do Paraná.

CDD 621.90294

TERMO DE APROVAÇÃO

JOSEF STANISLAW FALCÓN MAGALHÃES

SISTEMA VARIANTE PARA SELEÇÃO DE MÁQUINAS- FERRAMENTA

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Curso de Pós-Graduação em Engenharia Mecânica da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientador: Prof. Dr. Dalberto Dias da Costa
Departamento de Engenharia Mecânica - UFPR

Prof. Dr. Marcos Valério Ribeiro
Faculdade de Engenharia de Guaratinguetá - UNESP

Professora Dra. Maria Lúcia Leite Ribeiro Okimoto
Departamento de Engenharia Mecânica - UFPR

Curitiba, 09 de abril de 2008.

AGRADECIMENTOS

A Deus, pela minha vida, benção e proteção.

Ao professor Dalberto Dias da Costa pela íntegra orientação, principalmente pela compreensão e idéias.

Aos professores da UFPR, pelo apoio didático e disponibilidade.

Aos colegas de mestrado (F. Lajarin, J. Gabardo, A. Dobiguinies, E. Takano, G. Bazzi, R. R. Gouveia, J. Walter Bruno Filho) e graduação pela compreensão e apoio contínuos.

Aos prof. Fred L. Amorim e Osiris Canciglieri Jr. Da Pontifícia Universidade Católica e Prof. Hélio Pedrini da UFPR pela colaboração e disponibilidade.

Funcionários diretos e indiretos da UFPR (M. B. Tenório, Sr. Lucas Chaves, D. Maria A. Santos)

Empresas participantes do cadastro de máquinas pela atenção e disponibilidade, especialmente a Usimold Ltda, (Cida, Benedito) e Empresa Júnior Eng. Mecânica (COEM-Jr).

Aos Familiares, pelo contínuo e interminável apoio e compreensão.

À Capes, pelo apoio financeiro.

E a todos aqueles que colaboraram indiretamente de alguma forma.

RESUMO

A seleção automática de máquinas-ferramenta ainda é um problema não resolvido. Apesar do avanço das tecnologias de manufatura auxiliadas por computador a intervenção humana ainda é necessária. Em muitos casos, esses sistemas são modulares e dedicados a processos específicos de manufatura, tais como o fresamento, o torneamento, a injeção plástica, etc. A escolha de um destes módulos fica sempre a cargo de um planejador humano (processista). Em muito essas tecnologias têm evoluído tentando facilitar e agilizar o ciclo de produção. Porém estão sempre voltadas para grandes sistemas de produção e requerem alto custo de implementação. A seleção de insumos e equipamentos disponíveis de forma correta, e em curto intervalo de tempo, é um grande problema na manufatura. O objetivo deste trabalho é apresentar uma metodologia que contemple a problemática da seleção de máquinas-ferramenta. A estratégia principal adotada nesta dissertação foi baseada em regras de produção (tipo “IFTHEN” – “SE-ENTÃO”) formatadas como cláusula SQL (*Structured Query Language*) para integrar duas bases de dados. A primeira base de dados contém a representação geométrica parametrizada de um grupo de *features* de usinagem entre outros atributos. A segunda engloba a descrição geométrica e cinemática de máquinas-ferramenta e características particulares relevantes à sua seleção. As regras estão divididas em regras fixas, utilizadas sem modificações, e regras variáveis, adaptadas pelo usuário para restringir a busca. Um protótipo computacional, baseado na metodologia proposta, foi implementado contendo duas interfaces: uma para o cadastro de máquinas-ferramenta, *features* e regras, e a outra para dimensionamento de blocos e *features* em tempo de execução. Além disso, foi implementado também um mecanismo de inferência para realização desta busca. Para fins de teste do protótipo, utilizou-se uma base de *features* baseada na norma ISO 14649. A principal conclusão foi que o uso de regras de produção embutidas em cláusulas SQL são ferramentas flexíveis de programação que permitem a implementação de atualização de um sistema variante para a seleção de máquinas-ferramenta. E também que a utilização de *features* de manufatura normalizadas pode contribuir para facilitar o trabalho de implementação de sistemas variantes.

Palavras-chave: CAPP variante. Seleção de máquinas-ferramenta. Base de dados de máquinas-ferramentas.

ABSTRACT

The automatic selection of machine-tools remains an unsolved problem. Despite of the up-to-date CAM technologies, the human intervention is still necessary. In many cases, these systems are modular and dedicated to a specific manufacturing process, such as milling, turning, mold injection and so on. The previous choice of one of these modules is always under the responsibility of a human planner (process planner). In many cases this technology has been developing for big companies and demands high implementation cost. The selection of consumables and equipment, including the machine tools, in a short time is great deal in manufacturing systems. The purpose of this work is to present a methodology to face the problem of machine-tool selection. The main strategy adopted here was based on production rules ("IF-THEN") formatted as SQL clauses to integrate two bases of relational data. The first base of data contains a parametric representation of a group of machining *features*. The second one is related to the geometric and kinematics of machine-tools and some important characteristics necessary to their selection. The rules are divided in fixed rules, used without modifications during the search of machines-tool and variable rules, which are user adapted during the search. A human-machine interface was created for allowing the process specialist to write rules and insert machine tools and *features*. The *feature* dimensioning can be done in a dedicated interface to start the inference. The main conclusion it was that the use of built-in production rules in SQL clauses are flexible programming tools that allow the implementation and updating of a variant CAPP system for machine-tool selection. In addition, it was observed that the adoption of standard machining *features* can facilitate the implementation of variant systems.

Keywords: Variant CAPP. Machine-tool Selection. Machine tool Database.

LISTA DE ILUSTRAÇÕES

ILUSTRAÇÃO 1 - Exemplo da cláusula "UPDATE" do SQL	22
ILUSTRAÇÃO 2 - Exemplo da sentença "SELECT" da linguagem SQL	23
ILUSTRAÇÃO 3 - Exemplos de <i>Features</i> de Torneamento (Adaptado ISO 14649 P12)	26
ILUSTRAÇÃO 4 - Exemplo das <i>features</i> de usinagem (adaptado ISO 14649 P11)	27
ILUSTRAÇÃO 5 - Exemplo de atributos de máquinas-ferramenta	28
ILUSTRAÇÃO 6 - Exemplo de modelo de objetos da base de dados de Máquinas (adaptado ASHBY, 2004)	29
ILUSTRAÇÃO 7 - Roscamento cônico (adaptado ACHERKAN <i>et al.</i> , 2000).....	30
ILUSTRAÇÃO 8 - Posição relativa dos eixos segundo a nomenclatura formal	31
ILUSTRAÇÃO 9 - Esboço do diagrama de caso de uso (UML2) da proposta do sistema	35
ILUSTRAÇÃO 10 - Diagrama de tempos	36
ILUSTRAÇÃO 11 - Representação esquemática dos dados e relacionamentos.....	37
ILUSTRAÇÃO 12 - Divisão dos atributos das <i>features</i> de manufatura	38
ILUSTRAÇÃO 13 - Exemplo de visualização da cinemática da máquina-ferramenta necessária à usinagem da <i>feature</i>	40
ILUSTRAÇÃO 14 - Modelo de representação orientada a objetos de máquinas-ferramenta .	41
ILUSTRAÇÃO 15 - Fluxograma de criação das regras de produção.....	47
ILUSTRAÇÃO 16 - Adição de uma nova <i>feature</i>	48
ILUSTRAÇÃO 17 - Adição de uma nova máquina-ferramenta.....	48
ILUSTRAÇÃO 18 - Adição de uma regra a <i>feature</i>	49
ILUSTRAÇÃO 19 - Conversão de cadeia direta em cadeia reversa.....	49
ILUSTRAÇÃO 20 - Detalhe da Conversão das Regras de Cadeia Direta em Cadeia Reversa (SQL)	50
ILUSTRAÇÃO 21 - Algoritmo de separação das regras de produção	51
ILUSTRAÇÃO 22 - Fluxograma de busca de máquinas através de <i>features</i>	52
ILUSTRAÇÃO 23 - Interface de inserção da cinemática da <i>feature</i>	56
ILUSTRAÇÃO 24 - Inserção de cotas parametrizadas.....	57
ILUSTRAÇÃO 25 - Edição de máquinas-ferramenta e cinemática.....	58
ILUSTRAÇÃO 26 - Visualização das regras de produção.....	59
ILUSTRAÇÃO 27 - Cadastro de uma regra ("calculadora lógica")	59
ILUSTRAÇÃO 28 - Estudo de caso, criação de uma <i>feature</i> de fresamento.....	60
ILUSTRAÇÃO 29 - Telas do software na seqüência de seleção de <i>features</i> de fresamento. .	61
ILUSTRAÇÃO 30 - Diagrama de Classes do UML2, "logical view"	63

LISTA DE QUADROS

QUADRO 1 - Métodos avançados de busca (Adaptado Raphael e Smith, 2003, p-179-180).....	24
QUADRO 2 - Atributos de uma <i>feature</i>	38
QUADRO 3 - Dados de exemplo (máquinas utilizadas)	54
QUADRO 4 - Dados utilizados, definição de uma das máquinas utilizadas	54

LISTA DE SIGLAS

ABIMAQ	- Associação Brasileira da Indústria de Máquinas e Equipamentos
ABNT	- Associação Brasileira de Normas Técnicas
AP	- <i>Applications</i> - Aplicações
CAD	- <i>Computer Aided Design</i> - Desenho auxiliado por Computador
CAE	- <i>Computer Aided Engineering</i> - Engenharia auxiliada por Computador
CAL	- <i>Computer Aided Logistics</i> - Logística auxiliada por Computador
CAM	- <i>Computer Aided Manufacturing</i> - Manufatura auxiliada por Computador
CAP	- <i>Computer Aided Planning</i> - Planejamento Auxiliado por Computador
CAPP	- <i>Computer Aided Process Planning</i> - Planejamento de Processos Auxiliado por Computador
CAQ	- <i>Computer Aided Quality</i> - Qualidade Auxiliado por Computador
CAX	- <i>Computer Aided</i> - Tecnologias Auxiliadas por Computador
CE	- Caixa Envolória
CEP	- Controle Estatístico de Processos
CNC	- Comando numérico computadorizado
DAS	- <i>Data/Digital Automation Systems</i> - Sistemas de Automação Digital
DCS	- <i>Digital/Data Control System</i> - Sistema de Controle de Dados/Digital
DNC	- <i>Distributed/Direct Numerical Control</i> - Controle Numérico distribuído/direto
EDM	- <i>Electrical Discharge Machining</i> - Usinagem por Descarga Elétrica
EIA	- <i>Electronic Industries Alliance</i> - Aliança Industrial Eletrônica
ERP	- <i>Environment Resource Planing</i> - Planejamento de recursos
FMC	- <i>Flexible Manufacturing Cell</i> - Célula de Manufatura Flexível
FMS	- <i>Flexible Manufacturing System</i> - Sistema de Manufatura Flexível
GDL	- Graus de Liberdade
IEC	- <i>International Electrotechnical Commission</i> - Comissão Eletrotécnica Internacional
ISO	- International Standard Organization - Organização Internacional de Padronizações
JIT	- <i>Just-in-Time</i> - "Por demanda"
KS	- <i>Knowledge Systems</i> - Sistemas de Conhecimento
MRP	- <i>Manufacturing Resource Planing</i> - Planejamento de recursos de Manufatura
NBR	- Norma Brasileira
NC	- <i>Numerical Command</i> - Comando Numérico
OP	- <i>Operation Plan</i> - Plano de operações
PCP	- Planejamento e Controle da Produção
PLC	- <i>Product Life Cicle/Programmable Logic Control</i> - Ciclo de vida de Produto / Controle Lógico Programável
SQL	- <i>Structured Query Language</i> - Linguagem de Busca Estruturada
STEP	- <i>Standard for the Exchange of Product model data</i> - Padrão para Troca de dados de modelo de Produto
TC	- <i>Technical Cometeet</i> - Comitê Técnico
TG	- Tecnologia de Grupo
UML	- <i>Unified Modeling Language</i> - Linguagem de Modelamento Unificada
VRML	- <i>Virtual Reality Meta Language</i> - Meta-linguagem de realidade Virtual

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivo	11
2	PLANEJAMENTO DE PROCESSOS.....	12
2.1	Planejamento de Processos Auxiliado por Computador	13
2.2	<i>Features</i> de Usinagem.....	24
2.3	Máquinas-Ferramenta.....	28
2.4	Importância da Interface com o Usuário	32
2.5	Revisão Geral	33
3	METODOLOGIA	34
3.1	Escopo e Estrutura.....	34
3.2	Agentes.....	36
3.3	Modelo para Representação dos Dados	37
3.4	Fluxogramas de Atividades	46
4	IMPLEMENTAÇÃO, RESULTADOS E DISCUSSÕES	54
4.1	Validação	54
5	CONCLUSÃO	64
5.1	Sugestões de Trabalhos Futuros	65
6	REFERÊNCIAS BIBLIOGRÁFICAS.....	66

1 INTRODUÇÃO

O planejamento de processos é a ligação entre o projeto, o planejamento da produção e o chão de fábrica. As funções do planejamento são, por exemplo, selecionar e definir processos a serem executados em uma peça de maneira econômica, de acordo com as especificações de projeto, recursos disponíveis e demanda (tamanho de lote e prazos de entrega).

O planejamento de processo é uma atividade que depende largamente da experiência de processistas humanos. Devido a esta dependência, vários pesquisadores, por exemplo, Chang e Wysk (2002), van Houten (1991) vêm propondo o desenvolvimento de sistemas planejamento de processos auxiliado por computador (CAPP). A justificativa desses sistemas se encontra em eliminar a inconsistência de planos obtidos por processistas diferentes e melhorar a qualidade da documentação enviada ao chão de fábrica, garantindo-se, assim, o domínio do processo. Na literatura especializada, por exemplo, Alting, Zhang (1989), encontram-se dois tipos de abordagem diferentes para construção de um CAPP: planejamento variante; planejamento generativo automático.

Dentro desse panorama, a seleção de máquinas-ferramenta é uma atividade de suma importância, porém tem sido pouco estudada. Sendo o principal problema escolher as máquinas-ferramenta capazes de produzir um determinado produto a partir do seu modelo. Esse problema pode ser expandido caso o planejador tenha que selecionar máquinas em um acervo muito grande, por exemplo, disponíveis em uma rede local de fornecedores ou até mesmo desconhecidas para o planejador. E também tendo em vista novas máquinas multitarefas que não seguem uma nomenclatura formal.

Este trabalho se justifica quando a seleção de máquinas-ferramenta se torna um meio pelo qual um planejador pode encontrar uma máquina, dentro de um grande parque como grupo de empresas prestadoras de serviços. De um modo geral, pode-se considerar que a máquina-ferramenta seria capaz de prover o serviço, pois a ela em alguns casos pode representar o item mais dispendioso de um processo, quando comparados com outros recursos tais como dispositivos de fixação e ferramentas de corte.

Esta introdução contextualizou, de forma geral, o planejamento de processos, e sua importância. Apresenta-se, a seguir, uma revisão sobre seleção de máquinas-ferramenta no contexto do planejamento de processos. A metodologia para o desenvolvimento do sistema pretendido é apresentada no terceiro capítulo. Os detalhes de uma implementação computacional são apresentados no quarto capítulo, seguido pela conclusão e referências utilizadas neste trabalho.

Neste escopo, a proposição a ser apresentada e defendida neste trabalho é que existe pelo menos uma classe de máquinas cuja cinemática seja suficiente para usinar uma dada *feature*. Uma vez identificada essa classe, deve-se passar à identificação de qual modelo de máquina, dentro dessa classe, atende às exigências impostas por uma *feature* em particular, diferenciada por suas dimensões e tolerâncias.

1.1 Objetivo

O objetivo deste trabalho é apresentar uma metodologia para seleção de máquinas-ferramenta baseada em características geométricas da peça e, também, geométricas e cinemáticas das máquinas-ferramenta.

2 PLANEJAMENTO DE PROCESSOS

O planejamento é o argumento da ação. Todos os dias, para cada produto novo, um novo plano é compilado. Porém, em alguns casos, a experiência adquirida é tão valiosa quanto o plano. Simplesmente agir é um reflexo, mas o custo de investir em um planejamento prévio pode ser considerado, quando novos paradigmas são encontrados. (GHALLAB, 2004).

Planejamento de manufatura, engenharia de processos e seleção de máquinas-ferramenta são alguns poucos tópicos relacionados com o planejamento de processos (ALTING, ZHANG, 1989; CHANG, WYSK, 2002). Para a produção de um componente é necessário conhecimento de todo seu ciclo de vida. A transformação da matéria-prima em uma peça ou componente é denominada de manufatura, que hoje, apesar do nome, refere-se a um processo industrial. Em especial, processos de fabricação são modificações sucessivas que esta matéria-prima, normalmente um bloco de material ou pré-forma, se sujeita até efetivamente assumir todas as exigências geométricas e tecnológicas definidas.

Muitos autores (LEPIKSON, 2005, p.13-34, GHALLAB, et al. 2004, SRINIVASAN, SHENG, 1999) separam o planejamento de processos em macro e micro planejamento. A seleção de máquinas-ferramenta, por exemplo, em muitos casos tem sido uma atividade de macro planejamento (SRINIVASAN, 1999).

O planejamento de processos se inicia com a interpretação do modelo do produto, o qual deve conter informações geométricas e tecnológicas, tais como: material, tolerâncias e demanda (quantidade e prazo de entrega). Após essa interpretação definem-se, com poucos detalhes, os processos, máquinas e/ou equipamentos e o encadeamento dos mesmos. Entretanto, devido a esse baixo nível de detalhamento, ainda não se pode obter um roteiro de fabricação suficiente para dar subsídios a outros componentes do sistema de manufatura, tais como o planejamento e controle da produção (PCP).

Diante disso, faz-se necessário um refinamento do plano até então obtido, sendo acrescentadas informações mais detalhadas, as quais propiciarão descrever todas as ações necessárias à fabricação do referido produto (ALTING e ZHANG, 1989).

O plano assim obtido é, às vezes, denominado folha de operações ou folha guia. Nos sistemas de produção atuais, os planos são elaborados por um, ou mais, profissionais (processistas), cuja experiência é fundamental para o sucesso do plano. Isto torna o planejamento, mais uma arte do que um procedimento formal (CHANG e WYSK, 2002).

Em decorrência da grande disseminação de sistemas computacionais na manufatura, tais como o CAD e o CNC, e da escassez de processistas experientes, vários pesquisadores, dentre os quais se destacam: sistema XPLAN, sistema ACPSAP e sistema PART (ALTING, ZHANG, 1989; CHANG, WYSK, 2002, HOUTEN, 1991) vêm defendendo o desenvolvimento de sistemas baseados em computadores para auxílio (ou realização) ao planejamento.

Planejamento de processo é uma tarefa que requer uma quantia significativa de tempo e experiência. Automatizar planejamento de processo é uma alternativa óbvia para resolver este problema. Muito se estuda sobre sua automatização, mas as soluções genéricas não resolvem o problema, e, por outro lado, soluções específicas não atendem a grande maioria. A solução buscada é chamada de planejamento de processos auxiliado por computador (CAPP), apresentada a seguir.

2.1 Planejamento de Processos Auxiliado por Computador

As pesquisas recentes no desenvolvimento CAPPs indicam a criação de sistemas cada vez menos dependentes da intervenção humana. Alguns autores (ALTING e CHANG, 1989, LEE, *et al.*, 2007) defendem a implementação de sistemas automáticos para planejamento, isto é, sem nenhuma participação humana. Automáticos ou não podem propiciar as seguintes vantagens:

- redução da dependência humana;
- redução no tempo de planejamento de processos;
- redução no planejamento de processos e custos de fabricação;
- criação de planos com maior reprodutibilidade
- criação de planos com maior repetibilidade;
- com conseqüente aumento da produtividade.

Basicamente foram utilizadas duas abordagens para desenvolvimento de CAPPs: o variante e o generativo. A abordagem variante usa procedimentos de recuperação de bibliotecas para encontrar planos padrões para componentes com características semelhantes. Os planos padrões são criados manualmente por planejadores de processo.

O planejamento de processo é imprescindível para ligar o projeto à manufatura. Só podem ser traduzidas informações de projeto em idioma industrial através do planejamento de processo. Hoje, ambos, projeto auxiliado por computador (CAD) e manufatura (CAM), foram implementados. Integrado, ou ligado, estas funções requerem o planejamento de processo automatizado (ZEID, 2005; RAPHAEL, SMITH, 2003).

Um planejamento de processos variante utiliza a semelhança entre componentes para recuperar planos de processo existentes. Um plano de processo que pode ser usado por uma família de componentes é chamado um plano padrão. Um plano padrão é armazenado permanentemente no banco de dados com um número familiar como sua chave. Não há nenhuma limitação ao detalhe que um plano padrão pode conter. Porém, tem que conter uma sucessão de passos de fabricação ou operações pelo menos. Quando um plano padrão é recuperado, certo grau de modificação é normalmente necessário para tornar útil o plano para um componente novo.

O método de recuperação e a lógica em sistemas variantes são dados através do agrupamento de peças em famílias. Podem ser identificados métodos industriais comuns para cada família. Esses métodos industriais comuns são representados por planos padrões, como explicado anteriormente.

O mecanismo de recuperação de um plano padrão está baseado em famílias de peças. Uma família é representada por uma matriz que agrupa e inclui todas as possíveis peças associadas. A estrutura desta matriz de famílias será discutida no capítulo 2.1.1.1 (Tipos de inferência). Em geral, sistemas de planejamento de processos variantes têm duas fases operacionais: uma fase preparatória e uma fase de produção (CHANG, 2002).

Trabalho preparatório é necessário quando se implementa um sistema variante pela primeira vez. Durante a fase preparatória, são codificados

componentes existentes, classificados, e subsequente agrupam-se em famílias. O primeiro passo é escolher um sistema de codificação apropriado.

O sistema de codificação tem que cobrir o espectro inteiro de partes produzido no chão de fábrica (BURBIDGE, 1971). Deve ser sem ambigüidades e fácil de entender.

Características especiais, também chamadas de *features*¹ de manufatura, que existem nas peças devem ser completamente identificadas pelo sistema de codificação (SHAH, 1991; DHANDE, 1995). Um sistema de codificação existente pode ser adotado e então pode ser modificado para um chão de fábrica específico. A codificação de componentes existentes pode ser uma tarefa árdua. Antes de ser feito, um estudo completo do inventário de desenhos e planos de processo têm que ser executado, de forma que um ordenamento da tarefa de codificação possa ser administrado. O pessoal envolvido na codificação tem que ter uma compreensão precisa do sistema de codificação. Eles têm que gerar código idêntico para o mesmo componente quando eles trabalharem independentemente (BURBIDGE, 1971).

Depois que a codificação é completada as famílias de peças podem ser formadas. São muitas as técnicas de se agrupar componentes, essas são discutidas em tópico de tecnologias de grupos (RAPHAEL, SMITH, 2003). Uma matriz ou árvore de famílias é construída então para cada conjunto de peça. Devido ao grande número de componentes envolvidos, um computador pode ser utilizado para ajudar a construir as matrizes de famílias de componentes.

O próximo passo é preparar planos de processos padrão para cada família de peças. Resumindo, no processo de planejamento um conjunto de operações padrão (planos OP - *workplans*), podem ser identificados. Um plano de operação contém uma sucessão de operações de manufatura que normalmente são executadas em uma estação de trabalho, ou célula. Um identificador, um código de OP, é nomeado para cada plano de OP. Um plano de processo padrão é escrito em termos de códigos de OP e planos OP. São armazenados planos padrões então em um banco de dados indexam-se através de matrizes ou árvores de famílias (CHANG e WYSK, 2002).

¹ O termo *feature* será adotado neste texto na sua forma original (sem tradução) como sinônimo de características geométricas de manufatura.

Em muitos sistemas, são armazenados, também, planos de processos individuais no banco de dados. Porém, são armazenados somente os planos para peças freqüentemente produzidas. A fase preparatória é um processo intensivo de mão-de-obra e, ainda, requer uma grande quantia de esforço de todos. Tudo que está preparado para um chão de fábrica específico só pode ser utilizado para essa mesma. A estrutura de sistema e software pode ser utilizada em qualquer chão de fábrica, mas o banco de dados deve estar exclusivamente preparado pelo grupo desta e para cada chão de fábrica.

A fase de produção acontece quando o sistema estiver pronto para uso: podem ser planejados componentes novos neste instante. O código é a entrada para uma rotina de procura de peça-família para encontrar a família a qual o componente pertence. Desse modo, o número da família é utilizado para recuperar um plano padrão. O planejador humano pode modificar o plano padrão para satisfazer o projeto do componente.

Para uma peça freqüentemente produzida, seria desejável executar a procura diretamente através do código correto. Neste caso, um plano de processo (não mais um plano padrão, ou seja, “instanciado”) para uma peça existente é restaurado. Também podem ser somadas algumas outras funções, como seleção de parâmetro e cálculos de tempo padrão, para fazer o sistema mais completo, ou até mesmo a localização espacial das máquinas-ferramenta para cada plano de operações (ESAWI, ASHBY, *et al.*, 1998).

A abordagem generativa é considerada mais avançada como também mais difícil de desenvolver. Em um sistema de planejamento de processo generativo, são gerados novos planos de processo automaticamente para componentes sem recorrer a planos existentes (CHANG, WYSK, 2002; CANGIOLIERI Jr., 1999).

Estruturalmente sistemas generativos são constituídos de um sistema de inferência e bases de conhecimento (ALTING e ZHANG, 1989).

2.1.1 Sistemas Generativos

Peklenik (1989) demonstra que através da topologia do processo e um sistema de decisão o GPPS é capaz de gerar planos de processos para diferentes componentes. Este sistema de decisão é a inferência baseada em conhecimento

armazenado como dados, que seqüencialmente é preenchido pelo modelo do componente na entrada de um novo plano.

Sistemas de bases de conhecimento, também chamados “*knowledge bases*” (KB), são utilizados de diversas formas para solucionar o paradigma da inferência (RUSSEL, NORVIG, 2003).

2.1.1.1 Tipos de inferência

Os sistemas de conhecimento (KS – *knowledge systems*) são utilizados para solucionar a união das capacidades de um engenheiro de processos e os modernos sistemas computacionais disponíveis.

Armazenar conhecimento pode parecer fácil com as novas bases de dados, porém estas bases ainda somente auxiliam a tomada de decisão, não empreendendo, portanto, toda a capacidade computacional. Atualmente se tem em mãos um grande volume de informações sobre o chão de fábrica, mas pouco conhecimento sobre o que fazer com todos os dados juntos, pois, ainda dependem da decisão humana.

Uma grande combinação de metodologias pode ser utilizada na representação de conhecimento tais como: sistemas de busca, aplicativos de base de dados, estrutura de orientação a objetos, e sistemas de visualização. O conhecimento necessita de uma sintaxe para sua representação, e está dividido em (CHANG, *et al.*, 2002; RAPHAEL, SMITH, 2003, KUMAR, SINGH, 2007):

- Causal: onde existe uma causa e conseqüência;
- Classificatório: classifica dentro de um grupo;
- Composicional: não classifica, mas agrupa;
- Empírico: obtido através de experiência anterior;
- Princípios físicos: pré-definido por um parâmetro;
- Semântico: utiliza-se de sintaxe, por exemplo; e
- Heurístico: regras práticas.

O conhecimento causal é muito importante para tomada de decisão. Muitos dos outros conhecimentos podem ser reescritos na forma causal, porém apresentando riscos de perda da relação reversa ou reescrever a mesma relação várias vezes.

Primeiros sistemas de auxílio à decisão foram desenvolvidos na década de 70 empregando dados e regras. Essa aproximação ainda é muito útil para sistemas pequenos e de domínio bem definido onde o conhecimento muda muito pouco.

Desta maneira, a transformação de regras de conhecimento deve ser representada em declarações do tipo “SE... ENTÃO”, porém, informações importantes podem ser perdidas. Em contrapartida, dependências indesejadas podem ser adicionadas como o armazenamento da regra inversa, mesmo sendo recíproca (ALLADA, 2001).

A regra possui duas partes: a parte “SE”, também conhecida como antecedente e a parte conseqüente “ENTÃO” (do inglês *IF... THEN*). O antecedente contém um conjunto de condições que usualmente podem ser combinadas utilizando operadores lógicos “E” e “OU”. O conseqüente, “ENTÃO”, possui os elementos que incluem ações a serem tomadas (CHANG e WYSK, 2002).

Para simples representação, as condições do antecedente são comparações (maior que, menor que, e igual que) de valores de variáveis específicas. Entretanto, uma representação mais geral foi desenvolvida em que o antecedente consiste em padrões de dados que representam condições genéricas sob as quais as regras são aplicadas.

As linguagens de padrão-correspondente permitem regras genéricas definidas como na REGRA 1, onde *Mf* é uma máquina-ferramenta e *MI* um vetor contendo uma lista de máquinas-ferramentas a ser comparada e “*n*” uma posição qualquer deste vetor.

SE \exists outra *Mf* maior que *MI*(*n*) ENTÃO *Mf* é a Maior.

REGRA 1 - Exemplo de regra de cadeia direta

Regras genéricas que manipulam conjuntos de dados são vastamente utilizadas no “garimpo de dados”, por exemplo: geração de relatórios, filtro de e-mails e outras aplicações. Existem duas maneiras básicas de uso de regras causais do tipo “*IF... THEN*”: encadeamento direto (encadeamento progressivo ou para frente) e encadeamento reverso (encadeamento regressivo ou para trás) (CHANG e WYSK, 2002; RAPHAEL, SMITH, 2003).

Em um planejamento de processo variante, os planos de processos são retirados de uma base de dados. A direção para o procedimento de planejamento não existe porque, simplesmente, os planos são conectados ao código. Entretanto, quando em um planejamento de processos generativo, os planos que são gerados, definem um estado inicial para ordenar a busca para o estado final (objeto). O caminho tomado (inicial -> final ou final->inicial) representa a seqüência do processo.

Por exemplo, o estado inicial é o material bruto (peça de trabalho) e o estado final é o projeto das qualidades finais.

O planejamento reverso usa um procedimento retrógrado. Assumindo que se tem um componente acabado, o objetivo é preenchê-lo com a forma não usinada. Cada processo de usinagem é considerado um processo de preenchimento. Um processo de furação pode preencher um buraco, e assim por diante. Um processo de roscamento é selecionado (ele reduz a superfície da rosca para o menor diâmetro do furo com uma superfície de desbaste). Uma furação é selecionada para preencher o furo (CHANG e WYSK, 2002).

Planejamento direto e o reverso são muitos similares; entretanto eles afetam significativamente a programação do sistema. Planejar cada processo pode ser caracterizado pela pré-condição da superfície a ser usinada e após condição da usinagem (seu resultado).

Para o planejamento progressivo deve-se conhecer a superfície sucessora antes de selecionar o processo, porque após condição do primeiro processo torna-se a pré-condição para o segundo processo. Por exemplo, quando seleciona uma furação para abrir uma rosca em um furo se conhece qual rosca será cortada. Entretanto a broca de desbaste do furo usa o menor diâmetro. Caso contrário se uma broca maior for escolhida nenhuma rosca será produzida.

O planejamento reverso elimina o problema condicional porque ele começa com a forma da superfície final e os processos são selecionados para satisfazer os requerimentos iniciais. A superfície transiente (superfície intermediária) produzida pelo processo de preenchimento é a pior pré-condição que um processo de usinagem pode aceitar (por exemplo, a profundidade de um fresamento de acabamento e assim por diante). Qualquer processo de preenchimento que satisfaça a superfície transiente pode ser selecionado como processo sucessor (CHANG e WYSK, 2002).

No planejamento direto a superfície objetivo sempre deve ser mantida mesmo que muitas operações devam ser executadas para garantir o resultado. De outro modo, o planejamento reverso, inicia com os requisitos finais (que ajuda a selecionar o processo predecessor) e procura condição inicial ou algo menos preciso (que é fácil de satisfazer).

Em um sistema de planejamento de processos generativo, a lógica de decisão o sistema é o centro do software e dirige o fluxo de controle de programa. A lógica de decisão determina como são selecionados um processo ou processos. A função principal da lógica de decisão é emparelhar as capacidades de processo com a especificação de projeto. Capacidades de processo podem ser descritas por expressões "SE... ENTÃO...". Tal expressão pode ser traduzida em declaração lógica em um programa de computação. Talvez o modo mais eficiente para traduzir estas expressões seja codificar expressões de capacidade de processo diretamente em uma linguagem de computador. Podem ser traduzidas informações em manuais ou tabelas de limite de processo usando uma linguagem de computador de alto-nível facilmente. Porém, tal programa pode ser muito longo e ineficiente. Até mesmo mais desvantajosa é a inflexibilidade (dificuldade de modificação) de tal software, esta inflexibilidade deixa códigos personalizados deste tipo virtualmente inúteis planejando em curso (CHANG e WYSK, 2002).

Podem ser usados vários métodos para descrever a estrutura de decisão de planejamento de processo. Os métodos de representação do conhecimento estão diretamente relacionados à lógica de decisão nestes sistemas. Os dados estáticos são a representação e o uso dinâmico dos dados se torna a lógica de decisão. Listam-se lógicas de decisão a seguir aplicadas para sistemas planejamento de processos:

1. árvores de decisão;
2. tabelas de decisão; e
3. inteligência artificial.

Esta lista não está completa. Porém, esta classificação forma um embasamento para discussão (CHANG e WYSK, 2002).

Como o nome propõe, bases de conhecimentos devem ser armazenadas em algum meio físico de fácil acesso. Conceitualmente prefere-se a utilização de bases de dados para armazenar conhecimento de engenharia. Abiteboul e Hull (1995)

descrevem várias maneiras de armazenar dados e ressaltam que a engenharia é quem mais utiliza estas técnicas.

2.1.2 Bases de dados

Entre vários tipos de bases de dados, as que mais se destacam na engenharia são as bases de objetos e bases relacionais. As bases de objetos possuem a mesma estrutura em árvore, onde cada objeto “herda” as características dos pais, porém a duplicação de dados é inevitável (RAMAKRISHINAN e GEHRKE, 2002; KUNG *et al.*, 1999).

No entanto, as bases relacionais possuem a característica de evitar a duplicação de dados, conectando vários dados de mesmo tipo e ligando campos por meio de associações diretas. A mescla de bases de objetos com bases relacionais cria as bases multidimensionais, que preenchem o maior gradiente possível de tipo de dados (SILBERSCHATZ, 1986; KUNG *et al.*, 1999). Os sistemas de busca (QUADRO 1) em bases relacionais se popularizaram por meio da SQL (*Structured Query Language*), essa se tornou norma em 1992 (ISO/IEC 9075 - RAMAKRISHINAN e GEHRKE, 2002; ABITEBOUL *et al.*, 1995) com várias revisões (atualmente ISO 9075 de 2006).

2.1.3 Estrutura da linguagem SQL

A linguagem SQL foi criada para gerenciar banco de dados, a idéia era criar um sistema de gerenciamento de bases de dados (SGBD) padronizado. A estrutura lógica básica do SQL é dividida em cláusulas. As principais funções do SQL englobam a manipulação de dados, definição dos tipos de dados e a principal e mais complexa: a busca de dados (RAMAKRISHNAN, GEHRKE, 2002).

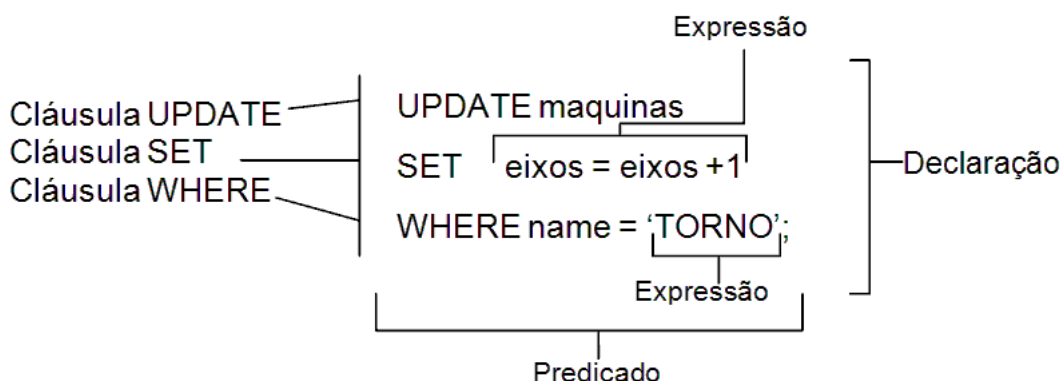


ILUSTRAÇÃO 1 - Exemplo da cláusula "UPDATE" do SQL

Na ILUSTRAÇÃO 1 observa-se um exemplo de cláusulas de manipulação de dados, onde o campo “eixos” da tabela “maquinas” é incrementado em 1, porém, somente onde o campo “name” é igual a “TORNO”.

A sentença mais comum, e a mais complexa, é a “SELECT”, utilizada para busca. Esta cláusula é seguida de uma lista de campos existentes em uma ou mais tabelas que é indicada na cláusula “FROM”. A cláusula “WHERE” é o predicado da sentença SQL, nela ocorre a qualificação dos dados, onde são selecionados segundo a condição lógica, imposta antecedida por esta cláusula (“WHERE”).

Na ILUSTRAÇÃO 2 constata-se que a condição imposta na cláusula “WHERE” é de igualdade. Podem-se considerar, além de expressões lógicas {<,<=,=,>,>=,>} convencionais, comparadores de texto {between,in,like} com curingas (*,%) que representam textos incompletos, operadores aritméticos {+,-,*,/,%}, e as novas correções (ISO 9075-14) de 2006 possibilitam operadores matemáticos trigonométricos (seno, cosseno e tangente). Esta flexibilidade torna a linguagem SQL versátil para implementações em qualquer ambiente computacional (RAMAKRISHNAN, GEHRKE, 2002; ABITEBOUL *et al.*, 1995; RAPHAEL, SMITH, 2003).

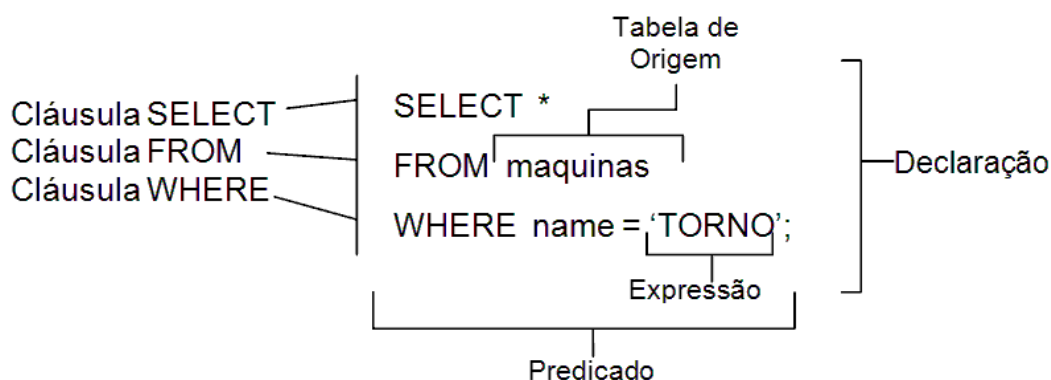


ILUSTRAÇÃO 2 - Exemplo da sentença "SELECT" da linguagem SQL

2.1.4 Sistemas de busca

A busca por uma máquina-ferramenta exige certas restrições. Em um estudo anterior (ASHBY *et al.*, 2004; KUMAR, SINGH, 2007), apresentaram uma comparação de softwares presentes no mercado capazes de efetuar, não só a seleção de processos, mas também materiais que são sistemas de busca baseados em três grupos básicos (CHTOUROU, *et al.*, 2005):

- Procura livre: uma procura é efetuada na base de dados por meio de comparação;
- Questionário: uma seleção de perguntas permite ao sistema iniciar uma procura;
- Raciocínio baseado em casos e Analogia: casos de conhecimento são estruturados e organizados em “árvores”, mede-se a distância do caso ao problema para encontrar o caso mais próximo.

No QUADRO 1 é apresentado um resumo das metodologias avançadas de busca em sistemas de engenharia, com características básicas e aplicações.

Essas metodologias podem ser aplicadas de forma individual, ou mescladas, para encontrar a solução da forma mais prática de representar máquinas-ferramenta e uma maneira compreensível de relacioná-las com a geometria do componente. Para tanto é necessária a representação do conhecimento (RAPHAEL, SMITH, 2003; ARSLAN, *et al.*, 2004).

Método	Melhor aplicação	Pontos Fortes	Pontos Fracos
Busca Exaustiva	Pequenas tarefas	Garantia de ótimo global	Complexidade Exponencial
Bifurcar e limitar	Variáveis discretas	Garantia de solução exata	Critério de ramificação e podas requeridas
Métodos gradientes	Funções matemáticas com poucos ótimos locais	Boa convergência para ótimo local	Torna-se cilada na ótima local
Busca em tabela (SQL)	Funções matemáticas com poucas variáveis	Fácil implementação	Solução identificada depende do tamanho da tabela
Busca linear	Funções matemáticas não-convexas com poucos ótimos locais	Eficiente	Solução identificada depende da escolha da direção de busca
Busca puramente aleatória	Extremamente grosseiro e domínios descontínuos	Simplicidade	Convergência Ruim
Simulação em Anel (SA)	Onde boas heurísticas podem ser utilizadas para movimento ponto-a-ponto	Funções de avaliação de caixa-preta podem ser usadas	Ótimo global pode não ser identificada
Algoritmos genéticos	Grandes problemas combinatórios	Funções de avaliação de caixa-preta podem ser usadas	Ótimo global pode não ser identificada
Busca aleatória controlada	Ausente de heurística e domínios descontínuos	Funções de avaliação de caixa-preta podem ser usadas	Convergência ruim
PGSL	Domínios razoavelmente suaves, variáveis contínuas.	Boa convergência, escalabilidade.	Variáveis discretas

QUADRO 1 – Métodos avançados de busca (Adaptado Raphael e Smith, 2003, p-179-180)

2.2 *Features* de Usinagem

Atualmente, a representação de um produto é um dos itens mais importantes para sua produção. O produto precisa ser detalhado ao nível de componentes para sua fabricação. As principais características de um componente mecânico para ser fabricado podem ser descritas como geométricas, topológicas, tecnológicas (DHANDE, *et al.*, 1995; RAPHAEL; SMITH, 2003). Os sistemas de codificação desenvolvidos em tecnologias de grupo são uma alternativa para a representação dessas características, porém o código gerado para um dado componente não é suficiente para criar uma representação inequívoca do mesmo (GAO e HUANG, 1996).

O termo “*feature*” não tem uma definição clara e exata, mas podem ser descritas como um conjunto de informações referentes aos aspectos de forma ou outros atributos de um componente (ALLADA, 2001). Gao e Huang (1996) propuseram o conceito de “*features* de forma”, classificando-as em: atômicas, primitivas e compostas. As primeiras incluem elementos geométricos básicos, tais como: pontos, linhas, arcos, planos e superfícies, os quais, quando reunidos darão origem as *features* primitivas. Ressaltos, rasgos, furos e bolsões, apresentados na ILUSTRAÇÃO 3 e ILUSTRAÇÃO 4, são exemplos de primitivas padronizadas. As *features* compostas são coleções de primitivas e/ou atômicas, as quais juntas podem executar uma função fornecida ou serem produzidas por operações similares.

Ainda segundo esses autores (ALLADA, 2001; GAO, HUANG, 1996), a representação por *features* não é suficiente para fins de planejamento de processos, requerendo informações adicionais que as contextualize em relação a um dado processo. Na usinagem, por exemplo, *features* podem ser vistas como volumes a serem removidos (no caso de cavidades) ou volumes remanescente (no caso de ressalto).

ISO 10303 é uma norma da ISO para a representação interpretável por computador da troca de dados de produto industrial. Seu título oficial é Sistemas de Automação Industriais e Integração - Representação de dados de Produto e Intercâmbio, e também é conhecido como STEP (*Standard for the Exchange of Product model data*) ou o Padrão para Troca de dados de modelo de Produto. O objetivo do padrão é prover um mecanismo que seja capaz de descrever dados de um produto ao longo do ciclo de vida, independente de qualquer sistema em particular. A natureza dessa descrição não só torna isto satisfatório na troca de arquivos neutros, mas, também, como uma base para implementar e compartilhar bancos de dados de produto e arquivos.

Tipicamente a STEP pode ser usado para trocar dados entre CAD, CAM (*computer aided manufacturing*), engenharia auxiliada por computador (CAE), Administração de Dados de Produto / EDM e outros sistemas de CAx. STEP engloba dados de produto e de projeto mecânico e elétrico, dimensionamento geométrico e tolerâncias, análise e fabricação, com informações adicionais específicas a várias indústrias como automobilística, aeroespacial, construção civil, naval, petróleo e gás, plantas de processo e outros.

Onde as partes 219, 223, 224, 238, 240 tratam especificamente de *features* da área de manufatura. Em especial a parte 224 possui uma classificação que faz referência a outra norma ISO, a ISO/IEC 14649, voltada para substituição da antiga norma ISO 6983 (Dados para comandos numéricos). Essa nova norma, também conhecida como STEP-NC, fornece modelos de dados para construção de comandos numéricos modernos.

Partindo do esquema de formação do plano de processos, a organização da estrutura de planos, da definição dos dados de usinagem e suas características geométricas (*features*). As *features* da STEP-NC formam um conjunto restrito de aplicações de usinagem. Além de definir o formato de dados, a norma é restringida ainda a três tipos de métodos de remoção muito conhecidos o torneamento, o fresamento e a eletro-erosão. Sendo que a última não foi totalmente oficializada. Porém, o torneamento (ISO 14649 - Part 12) e o fresamento (ISO 14649 - Part 11) estão completamente definidos e publicados. Apesar das restrições a norma visa melhorar e padronizar meios de construção de peças por usinagem dividindo em grupos geométricos conforme ILUSTRAÇÃO 3 e ILUSTRAÇÃO 4, a seguir (HAN, *et al.*, 2000).

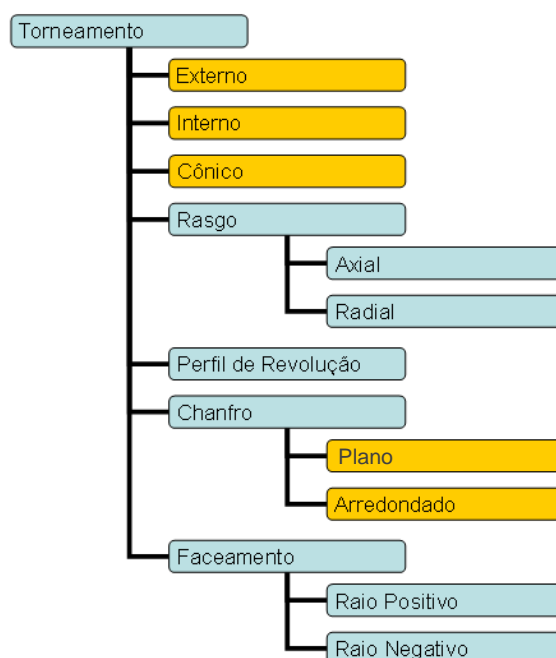


ILUSTRAÇÃO 3 - Exemplos de *features* de torneamento (adaptado ISO 14649 P12)

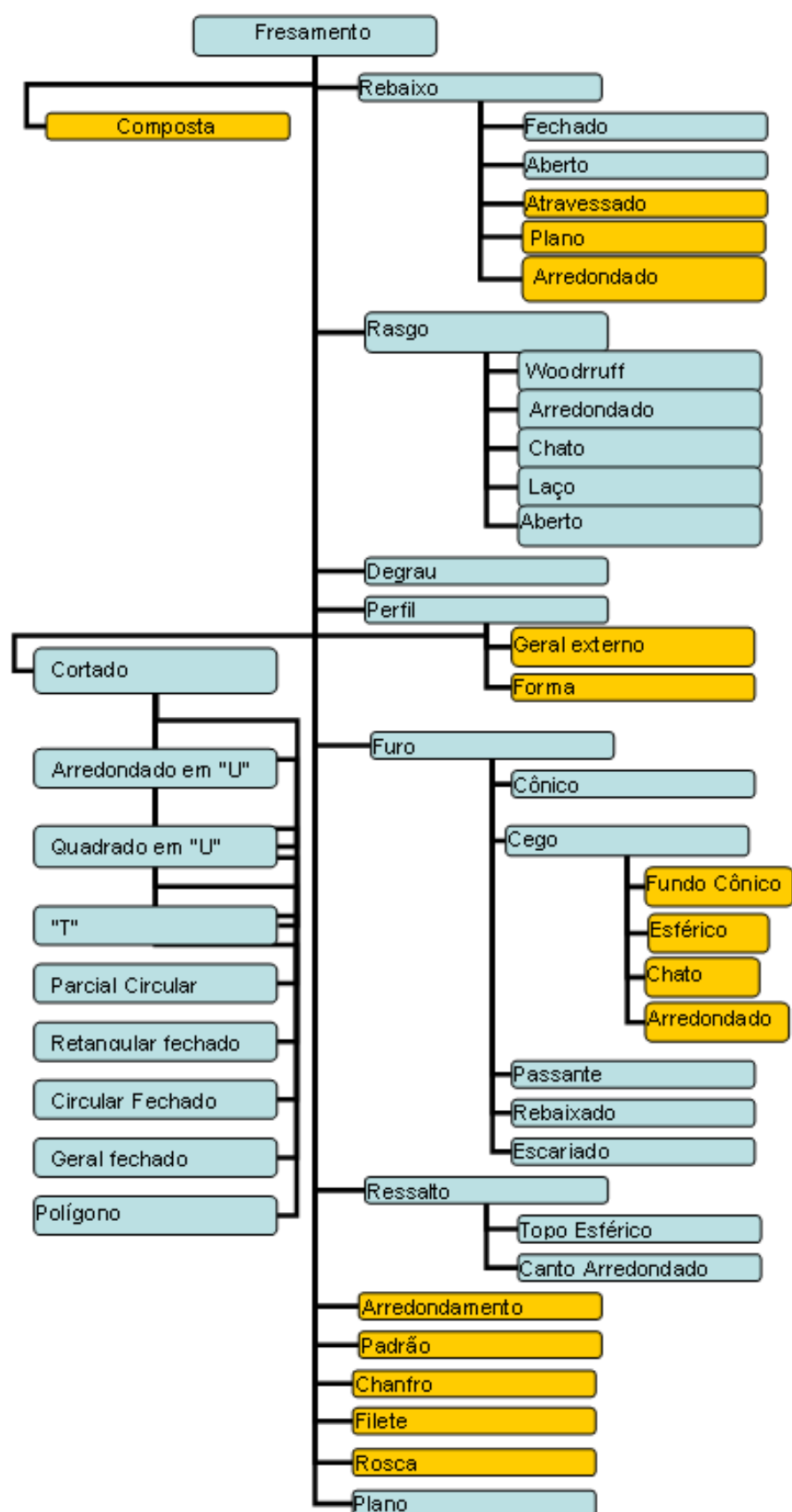


ILUSTRAÇÃO 4 - Exemplo das *features* de usinagem (adaptado ISO 14649 P11)

2.3 Máquinas-Ferramenta

Máquina-ferramenta define todo equipamento por meio do qual pode se operar uma ou mais ferramentas mecanicamente, com ou sem controle direto. As máquinas-ferramenta removem material da peça de maneira progressiva, até assumir a forma e tolerâncias desejadas. A maneira com que remove o material praticamente classifica a máquina-ferramenta (BLACK, 2002).

Do ponto de vista funcional as máquinas são divididas pelas operações que são capazes de efetuar. Algumas são concebidas para desempenhar apenas um tipo de operação, chamadas dedicadas, ao passo que outras compreendem um número maior de operações, denominadas multifuncional ou multitarefa (CHIN e CHEN, 2007).

Para a seleção de máquinas-ferramenta, existem alguns atributos, ou propriedades, apresentados na ILUSTRAÇÃO 5, que são mandatórios na escolha para produção de um determinado componente. Esses são presumíveis de se generalizar, porém, nem todos descrevem a máquina completamente. A ILUSTRAÇÃO 5 contém alguns exemplos tais como: P_{\max} o peso máximo admissível na mesa, Pow_{\max} representa a potência máxima do eixo árvore, X_{\max} e Y_{\max} é a representação do tamanho da mesa, e, θ_{\max} o ângulo máximo que bascula a mesa.

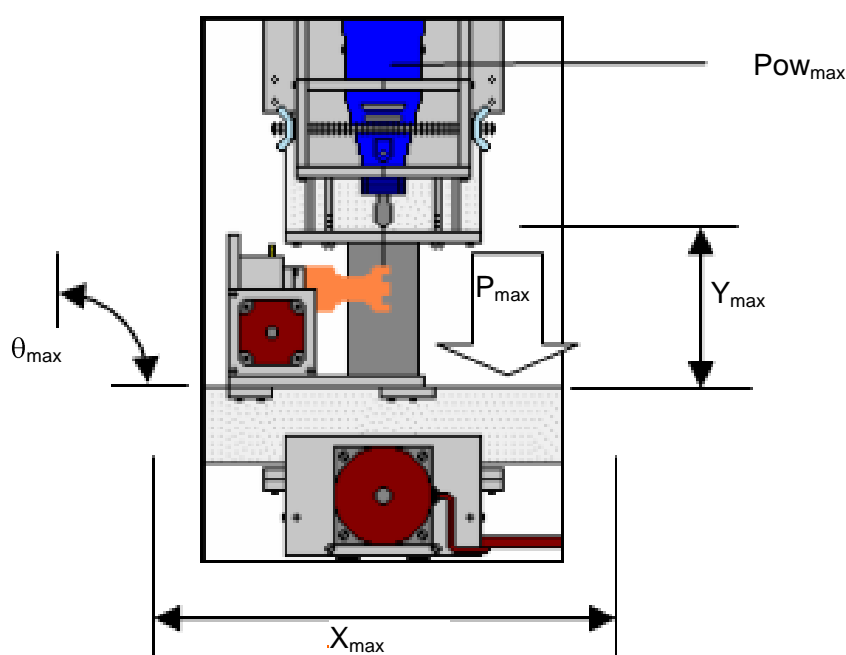


ILUSTRAÇÃO 5 - Exemplo de atributos de máquinas-ferramenta

Enumerar as propriedades necessárias para a seleção de uma máquina é uma tarefa difícil e necessita de um conhecedor profundo de cada máquina, o especialista.

2.3.1 Classificação

A metodologia de programação orientada a objetos mostra-se como grande ferramenta de representação abstrata do mundo real. Na engenharia de fabricação também se tem modelos abstratos, baseados em objetos, porém esta representação deve abranger todas as características necessárias para ligar as operações envolvidas.

É importante ressaltar que uma vez abstraídas máquinas e as *features*, estas relações não correspondem mais a vistas diretamente, como geométricas, podendo-se inclusive, obter relações complexas entre os objetos (SILBERSCHARTZ *et al.*, 1986).

Estas relações entre geometria e operação possuem intrinsecamente ligações com a máquina capaz de gerar uma superfície (ACHERKAN *et al.*, 2000). Estas relações podem ser representadas em um conjunto finito para um dado processo, como é o caso da usinagem, em que Dhande (1995) propôs um destes modelos. Porém os sistemas de busca se estendem além da geometria, exigindo sempre a presença do processista para inserir dados no sistema (SHAH, 1991).

Máquinas-ferramenta possuem várias propriedades e atributos específicos. Podem-se modelar máquinas-ferramenta em classes divididas, primeiramente, por tipo de máquinas, depois por subclasses de funções extras, e finalmente, nas instâncias destes objetos (máquinas-ferramenta) com suas devidas especificações, inclusive números identificadores de objetos (chave única e característica). ASHBY (2004, p.51-67) propôs este modelo em um de seus trabalhos, o qual pode ser visto no esquema da ILUSTRAÇÃO 6.

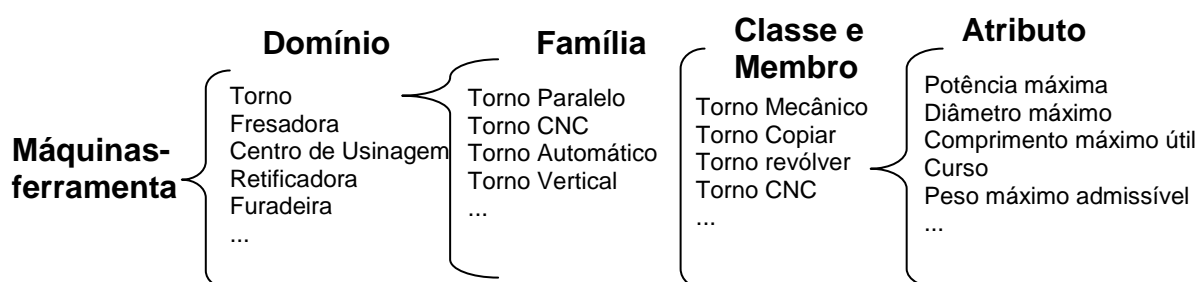


ILUSTRAÇÃO 6 - Exemplo de modelo de objetos da base de dados de Máquinas (adaptado ASHBY, 2004)

Outras classificações baseadas em nomenclaturas são de amplo conhecimento por parte dos especialistas. A indústria de máquinas-ferramenta utiliza como maneira de busca esta classificação (ABIMAQ). Contudo excluem-se as máquinas-ferramenta que possuem modificações, tornando sua função difícil de classificá-las, como por exemplo, as máquinas-ferramenta modernas multifuncionais. Sistemas mais adequados para representação computacional são baseados na representação cinemática, proposto por Acherkan *et al.* (2000), Peklenik (1989), Chin, Chen (2007).

2.3.2 Cinemática

Acherkan *et al.* (2000) propuseram uma semântica para representação dos processos de manufatura. Nesse trabalho uma superfície real é obtida pelo movimento relativo, denominado movimento formativo, entre duas linhas de geração (geratriz e diretriz). A geratriz e diretriz podem ser obtidas por quatro métodos distintos²: formação, geração, copiagem e tangente. De um modo geral a geratriz é obtida pelo movimento ou forma predeterminada da ferramenta. E a diretriz é o percurso por onde essa será percorrida. Além disso, Acherkan *et al.* ainda apresentam o conceito de movimentos operativos. Os quais são necessários para uma dada operação produtiva. Os movimentos operativos são compostos por um ou mais movimentos elementares: rotação e translação. Por exemplo, em uma operação de roscamento cônico (ILUSTRAÇÃO 7), os movimentos elementares de rotação (peça), translação linear (radial e axial) são necessários.

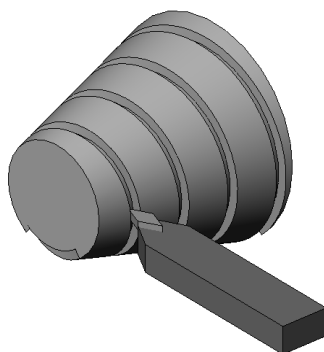


ILUSTRAÇÃO 7 - Roscamento cônico (adaptado ACHERKAN *et al.*, 2000)

² Tradução para os termos em inglês: *forming, generating, tracing, tangent.*

Apesar da taxonomia proposta por Acherkan *et al.*(2000) ser útil para a análise cinemática dos processos, sua representação computacional não é tão simples. Em um trabalho posterior Dhande *et al.* (1975) adaptaram a taxonomia proposta por Acherkan *et al.* (2000) possibilitando uma representação computacional por meio de uma modelagem matemática da mesma. Essa metodologia permite a determinação da superfície resultante após a determinação das duas linhas de geração (geratriz e diretriz). De acordo com esses autores isto permite entre outros a determinação da trajetória e detecção de colisões entre peça e ferramenta. Entretanto essa metodologia ainda não é suficiente para seleção da melhor máquina para executar uma determinada operação de usinagem.

A norma nacional NBR 6162 descreve os principais movimentos das máquinas-ferramenta, baseada na norma DIN 6580, dentre os quais se destacam os movimentos de: aproximação (da ferramenta até a peça antes da usinagem); corte (movimento em que ocorre a remoção de um único cavaco sem o avanço); e avanço (caminho pelo qual a ferramenta removendo cavacos continuamente até o término da operação).

Algumas normas (Norma ISO/R 841, EIA RS267-B) descrevem a nomenclatura dos eixos em máquinas de comando numérico seguindo a regra da mão-direita: polegar é o eixo X positivo, indicador o Y positivo e dedo do meio eixo Z positivo, tomando como referência a ferramenta de corte (OBERG, *et al.*,2004). Separando em movimentos lineares (somente um eixo), interpolados (dois ou mais eixos) conforme exemplo demonstrado na ILUSTRAÇÃO 8.

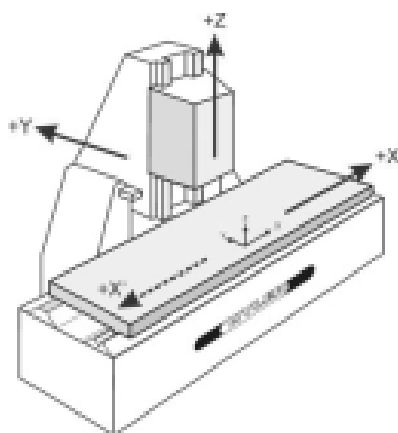


ILUSTRAÇÃO 8 - Posição relativa dos eixos segundo a nomenclatura formal

2.4 Importância da Interface com o Usuário

Para melhor compreensão, a modelagem de objetos especificada deve ser a de operações, pois estas permitem relações biunívocas e bijetoras, entres as *features* e as máquinas e que possam ser alteradas em qualquer instante. A um nível mais simplista: várias *features* apontam para várias máquinas. A ligação entre suas propriedades são funções compostas, o que justifica a criação do objeto *regra* (Sudarsan, Fenves, Siriram, 2005).

A modelagem destes objetos (*features* e máquinas-ferramenta) em uma base de dados, nem sempre está diretamente ligada às relações destes objetos. O nível de abstração das relações deve ser ainda maior, pois, a relação entre dados é diferente das relações entre objetos. Uma base de dados relacional liga os dados na representação e não denota o grau de relação entre os objetos (CHUNG e PENG, 2004; KUNG, *et al.*, 1999; TIAN, TAYLOR, 2002).

Atualmente para um processista, somente o conhecimento técnico não é mais suficiente para iniciar um planejamento de processo. Com as facilidades que o computador proporcionou o processista cada vez mais depende do computador para auxiliar no seu planejamento. Porém os computadores, como detentores dos dados para geração de conhecimento, podem adaptar-se a sistemas técnicos mais facilmente (FENG, 2005; RAPHAEL, SMITH, 2003).

A interface gráfica de um computador além de fornecer a informação necessária deve permitir usuários leigos aprenderem mais rapidamente. É o caso da utilização de tecnologias computacionais mais acessíveis para educar nas técnicas do planejamento de processos e manufatura (FENG, 2005; AVGOUSTINOV, 2000). Raphael e Smith (2003) enumeram as principais características de sistemas computacionais que tratam com conhecimento:

- Introduzir informações que são específicas da tarefa;
- Apresentar soluções múltiplas ao usuário em uma maneira compreensível;
- Permitir exploração de soluções dentro do espaço de soluções ótimas;
- Explicar o raciocínio, assim o usuário pode entender melhor porque foram propostas certas soluções e outros não;
- Permitir acrescentar conhecimento ao sistema;
- Ativamente "sugerir" ao usuário fazer boas escolhas sem tentar adivinhar o próximo movimento deles;

- Assinalar informações contraditórias;
- Avaliar combinações diferentes de escolhas;
- Ensinar os usuários como utilizar o sistema.

Utilizando os conceitos desta revisão introduz-se uma metodologia para resolver a problemática da seleção de máquinas-ferramenta.

2.5 Revisão Geral

Conforme os autores apresentados propuseram, as tecnologias de representação geral de dados de manufatura e elementos específicos evoluíram. Hoje, somente sistemas robustos utilizam a tecnologia de armazenagem de conhecimento para seus produtos (LIN e HSUA, 2008). Na usinagem, apesar de estar disponível em alguns softwares comerciais, como o CATIA V5, são pouco utilizadas.

3 METODOLOGIA

Como apresentado no capítulo anterior, os sistemas para planejamento denominados variantes são, em grande parte, baseados em tecnologia de grupo e atuam estritamente como sistemas de apoio à decisão. Nesta proposta, objetiva-se também a construção de um sistema para apoio à seleção de máquinas. Entretanto, ao invés de estar fundamentada nos conceitos da TG, a metodologia descrita a seguir baseia-se no uso de *features* de usinagem.

Apesar de parecerem conceitos muito diferentes, pois a TG é, em essência, o agrupamento de componentes similares e as *features* representarem o oposto, ou seja, a fragmentação dos componentes. Existem certas semelhanças entre TG e *features*, o que permite afirmar que a proposta deste trabalho possa ser considerada como um sistema variante para seleção de máquinas-ferramenta.

Esta possibilidade de fragmentação existente no planejamento baseado em *features* é inerente à usinagem, pois enquanto processo de remoção torna-se possível a fabricação de um único componente em diversas etapas, também denominadas “operações”. Dessa forma, defende-se aqui a existência de uma correspondência direta entre *features* e operações de usinagem.

Na atual definição apresentada nas normas STEP 10303 (224) e STEP-NC 14649 as *features* são apresentadas na forma parametrizada, isto é, o que importa é a geometria e a topologia, sendo que as cotas variam de acordo com dimensionamento do componente, dentro da faixa estipulada durante a criação desta. Dessa forma, torna-se razoável supor que o conceito de *features* representa um conjunto (família) de entidades geométricas com a mesma organização espacial (topologia), mas que podem ser dissimilares em dimensões, o que praticamente se assemelha ao conceito de TG.

3.1 Escopo e Estrutura

Na ILUSTRAÇÃO 9 existe uma representação esquemática da metodologia proposta. O que é denominado de camada de “modelagem” (ou preparação) refere-se à definição da inferência e estruturação do conhecimento e deve ser definida em

tempo de projeto, permanecendo imutável e independente da quantidade e qualidade do conhecimento a ser inserido.

A segunda camada é relativa ao tempo de manutenção e contém os recursos necessários à aquisição do conhecimento. Em outras palavras, ela contém a interface entre o sistema e o especialista humano, propiciando ao mesmo a inclusão de *features*, máquinas e regras de produção. Esse especialista é um processista com conhecimento técnico suficiente para classificar a *feature* e a máquina seguindo a estrutura de dados, tal como na camada de modelagem.

A interface com o usuário final é definida na terceira camada, a qual caracteriza o tempo de execução do sistema. É a partir dessa camada que o mecanismo de inferência é disparado, após o dimensionamento, pelo usuário, da *feature* a ser usinada. Este usuário pode ser leigo na seleção de máquinas, mas conhece a geometria da *feature* que deseja fabricar.

O conhecimento adquirido do especialista é armazenado em três bases: *features*, máquinas e regras. Essas são, em tempo de execução, gerenciadas pelo mecanismo de inferência, o qual ainda controla a interface com o usuário final.

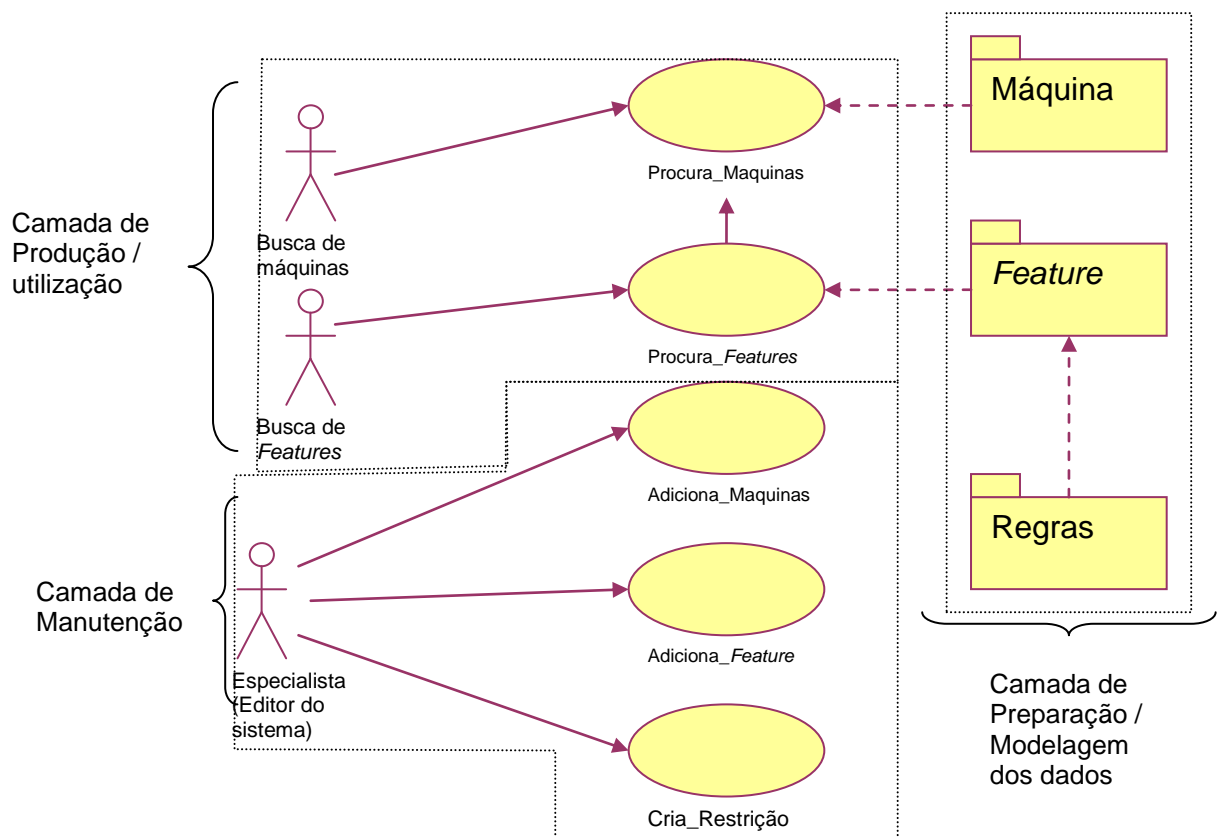


ILUSTRAÇÃO 9 – Esboço do diagrama de caso de uso (UML2) da proposta do sistema

3.2 Agentes

Para preenchimento da base de dados, além do conhecimento de um processista comum, se faz necessário conhecimento de sistemas computacionais. Porém, após esta estruturação, durante a etapa de “manutenção”, quando o sistema é continuamente alimentado um processista com conhecimento técnico na seleção de máquinas-ferramenta e mínimo de conhecimento de computadores deve preencher a base com os dados essenciais. Esse agente não necessita de conhecimento de regras complexas, somente regras do tipo “SE... ENTÃO”. Por este motivo, as regras em formatos exóticos para ele não devem ser exigidas.

Durante o preenchimento das bases processista classifica esta máquina através de uma classificação comum, além de preencher com detalhes da máquina-ferramenta. Deste modo, mais tarde, um processista pode executar uma busca dentro de uma classe específica de máquinas, se for necessário.

Uma vez iniciado o preenchimento das bases de dados, um segundo agente (usuário humano ou outro sistema) atua em tempo de execução na camada de busca, conforme a ILUSTRAÇÃO 10. Pressupondo que algumas *features* do chão de fábrica já tenham sido cadastradas, este usuário, (o qual pode ser um leigo na seleção de máquinas), pode executar uma busca na base de dados a fim de buscar a melhor máquina capaz de produzir este componente.

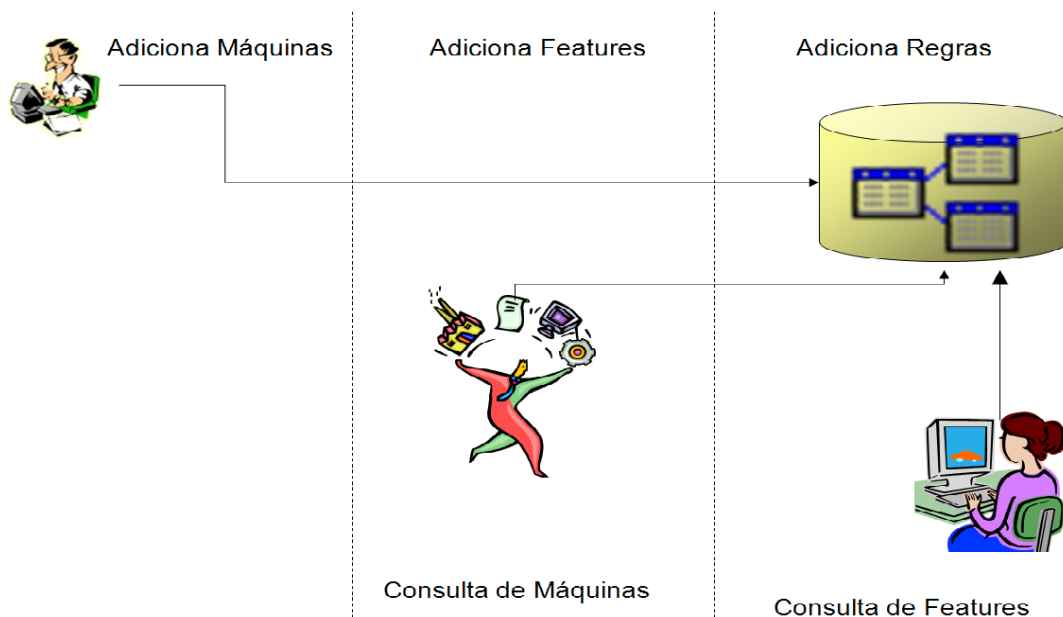


ILUSTRAÇÃO 10 - Diagrama de tempos

3.3 Modelo para Representação dos Dados

As bases de dados devem conter uma representação das *features*, máquinas-ferramenta e das regras. Cada uma delas se desmembra em um objeto com vários atributos, tal como descrito a seguir.

Uma visão geral da proposta da base de dados relacional pode ser observada na ILUSTRAÇÃO 11. Naturalmente, esta não contém as particularidades de cada máquina-ferramenta ainda. Porém, espera-se que com o contínuo preenchimento dos dados as bases de máquinas-ferramenta, *features* estendam-se em muitas outras tabelas ao passo que a árvore de dados também cresce.

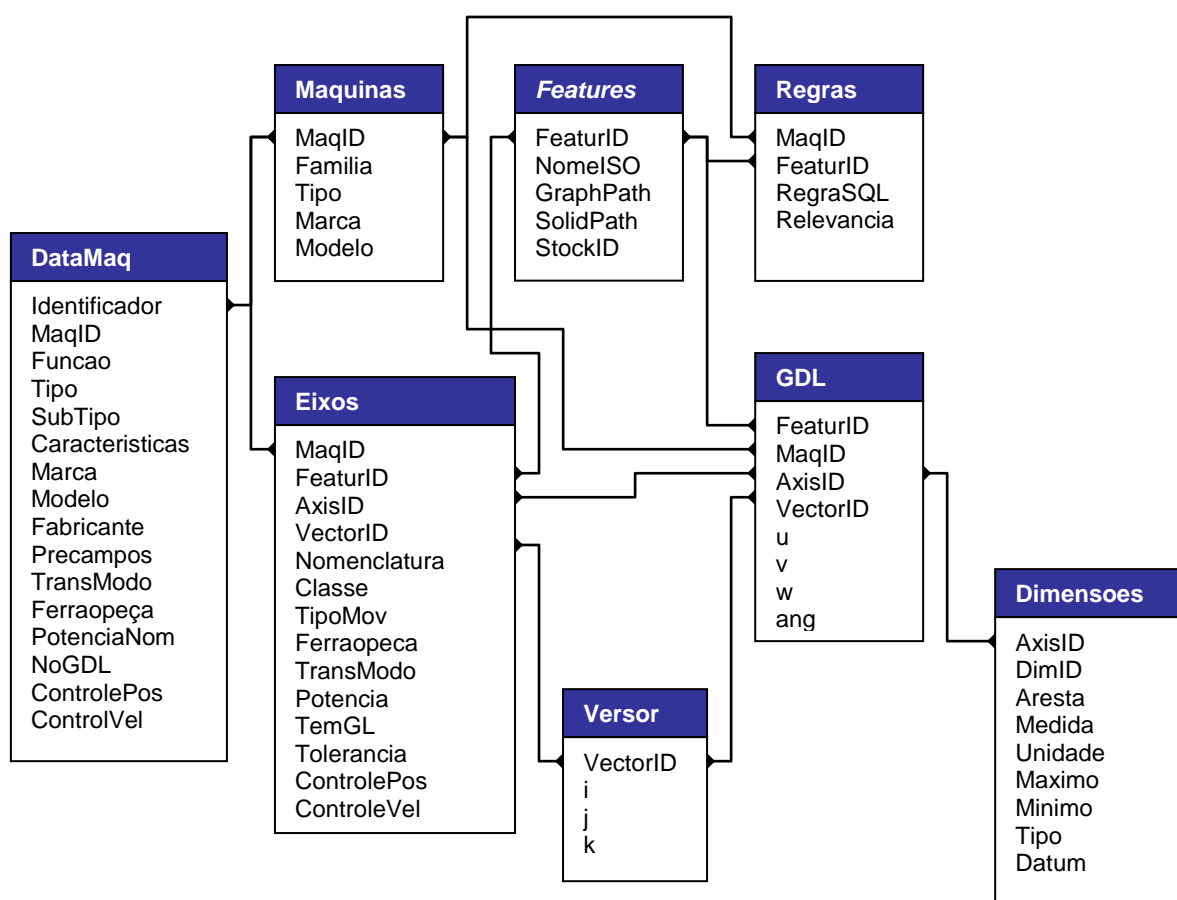


ILUSTRAÇÃO 11 - Representação esquemática dos dados e relacionamentos

3.3.1 Features

A representação das *features* pode ser tridimensional como entidade de geometria padronizada (baseada na norma ISO 14649). Sua utilização deve conter parâmetros necessários para determinação da operação e conseqüentemente a máquina.

Além da representação geométrica (aparência e dimensões) deve conter os seguintes aspectos topológicos: posição relativa e estrutura de sólidos, superfícies, arestas e vértices. Como, também, os aspectos tecnológicos: tolerâncias de medidas, sólido de origem e dimensões desses. Incluem-se, ainda, aspectos operacionais, enquadrados como aspectos tecnológicos, pois carregam a descrição cinemática da máquina, porém sendo atributo exclusivo da *feature*.

Enumeram-se no QUADRO 2, a seguir, as características de uma *feature* com uma representação destes atributos na ILUSTRAÇÃO 12.

Atributos geométricos	Geometria (pontos, arestas, faces, sólidos); Topologia (ordenamento, estrutura); Posicionamento espacial.
Atributos tecnológicos	Faixas de tolerâncias, material bruto de origem.
Atributos operacionais	Cinemática (eixos de posicionamento, avanço e corte seu posicionamento relativo e características).

QUADRO 2 - Atributos de uma *feature*

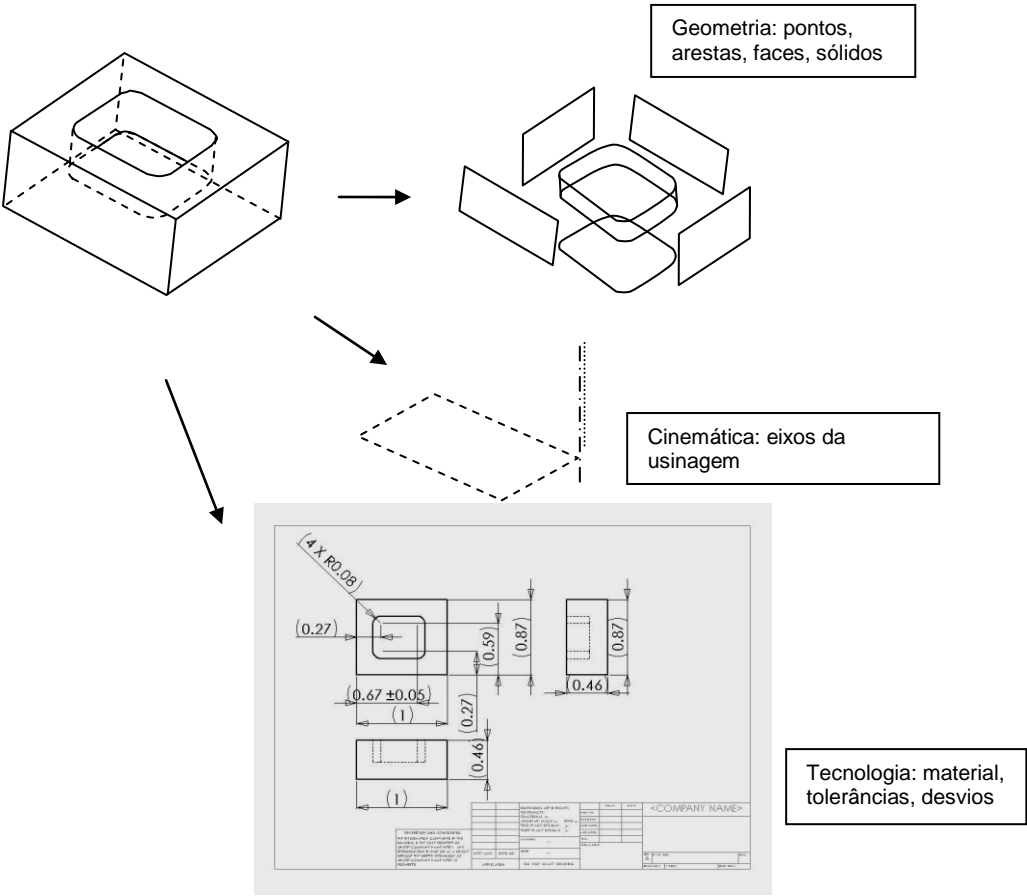


ILUSTRAÇÃO 12 - Divisão dos atributos das *features* de manufatura

A ligação entre *feature* e máquina-ferramenta já poderia ser determinada através do material bruto (stock), se houvesse ligação direta da máquina-ferramenta com a *feature*. Seu cadastro inicial deve ser feito de maneira gráfica, para que não se exija muito da parte computacional do processista, uma interface tridimensional permite inserir desenhos exportados de softwares CAD e manipular os atributos inseridos dentro da interface.

Concluída esta etapa, o processista personaliza a *feature* para um determinado grupo ou tipo de máquinas-ferramenta através das regras de produção. O ideal é que para cada *feature* exista apenas um determinado tipo de máquina. Como isso não é possível, então o especialista deve criar regras (nível hierárquico mais alto) para que durante a inferência, eles possam ser diferenciados.

3.3.1.1 Associação de cinemática

Procurou-se descrever máquinas-ferramenta através de sua cinemática. Conforme comentado anteriormente, são de comum conhecimento três movimentos básicos de uma máquina-ferramenta (seção 2.3.2): como posicionamento, avanço e corte. As máquinas-ferramenta, a priori, não estão agrupadas por nomenclaturas. A topologia do sistema de eixos cinemáticos oferece a possibilidade de agrupá-las.

Desse modo, máquinas de mesma cinemática estão agrupadas, porém podem não executar a mesma tarefa. Exemplo disso são tornos verticais e horizontais que possuem a mesma cinemática, mas difere quanto à aplicação e posicionamento relativo.

Na ILUSTRAÇÃO 13 exemplifica-se o caso de cinemática aplicada a uma *feature* de maneira visual e mostra também uma visualização da ferramenta para permitir compreender sua cinemática.

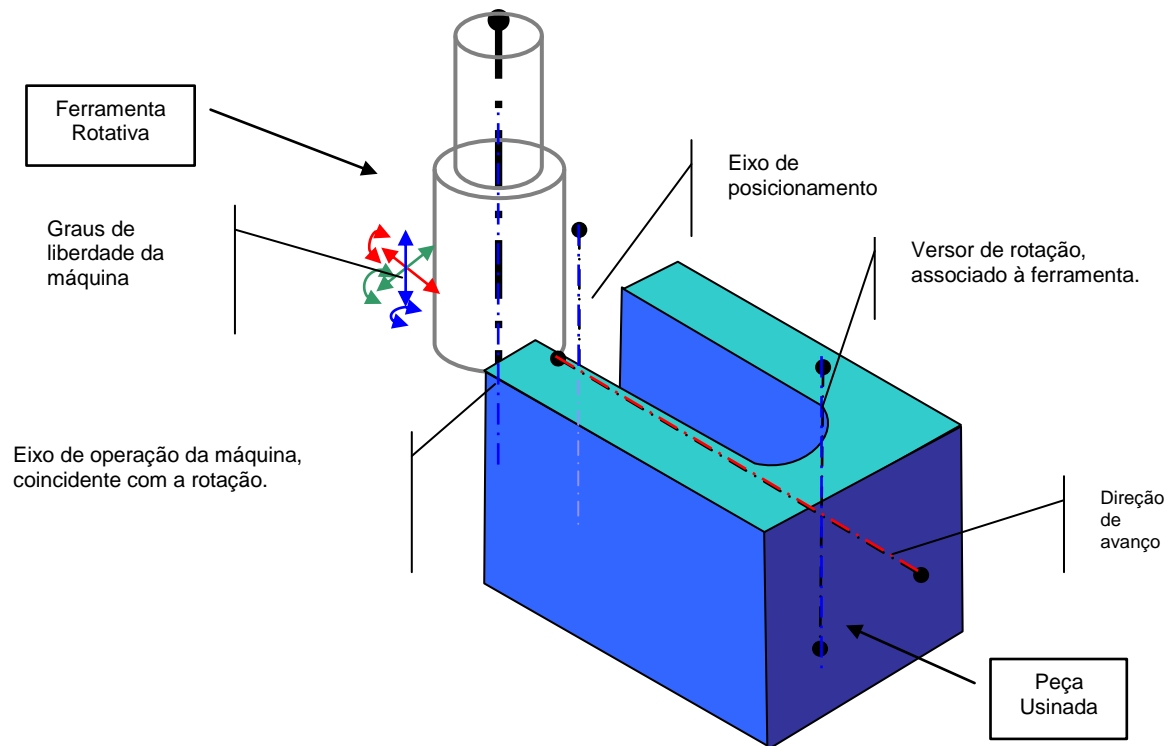


ILUSTRAÇÃO 13- Exemplo de visualização da cinemática da máquina-ferramenta necessária à usinagem da *feature*.

3.3.2 Máquinas-ferramenta

Para permitir a ligação, mesmo indiretamente, das *features* com as máquinas-ferramenta deve-se considerar: a estrutura orientada a objetos da classificação formal e comercial; máquinas multifuncionais; subutilização de máquinas dedicadas ou antigas e opções de refinamento como atributos adimensionais.

Enumeram-se as propriedades genéricas para definir uma máquina-ferramenta:

- Classificação formal;
- Rotação relativa da peça em relação à ferramenta;
- Potência por eixo;
- Tipo de movimento por eixo;
- Controle de velocidade por eixo; e
- Cinemática por eixo.

Essas propriedades são continuamente “herdadas” pelas máquinas conforme são definidas, ou seja, instanciadas.

3.3.2.1 Cinemática de máquinas-ferramenta

A cinemática mínima de uma máquina-ferramenta deve permitir pelo menos representar o eixo árvore, ou eixo principal desta máquina. Deve fazer parte do conhecimento do processista normas de nomenclatura, classificação e posicionamento dos eixos de movimento e rotação de uma máquina-ferramenta.

Somente a cinemática não é condição suficiente para representar a máquina. Pois duas máquinas com a mesma cinemática podem não executar a mesma *feature*. A máquina também precisa de parâmetros específicos impostos pelo especialista:

- Família, tipo, marca e modelo (nomenclatura comercial para classificação e busca da máquina) segundo ABIMAQ (ABIMAQ, 2008) conforme ILUSTRAÇÃO 14;
- Especificação técnica da máquina: número de graus de liberdade (GDL), número de eixos (x, y, z, w, etc.), opcionalmente pode-se informar se a rotação (caso exista) esteja no eixo de corte ou na peça;
- Especificação de cada eixo de trabalho, principalmente eixo árvore: tipo de movimento de cada eixo (posicionamento, avanço e corte), modo de translação (retilíneo, interpolado), controle de posicionamento (sem posicionamento, automático, manual, eletrônico) e comprimento de trabalho (tamanho da mesa), controle de velocidade (sem, manual, automático, eletrônico), e campos extras como potência nominal do eixo, tolerância e/ou resolução de trabalho.

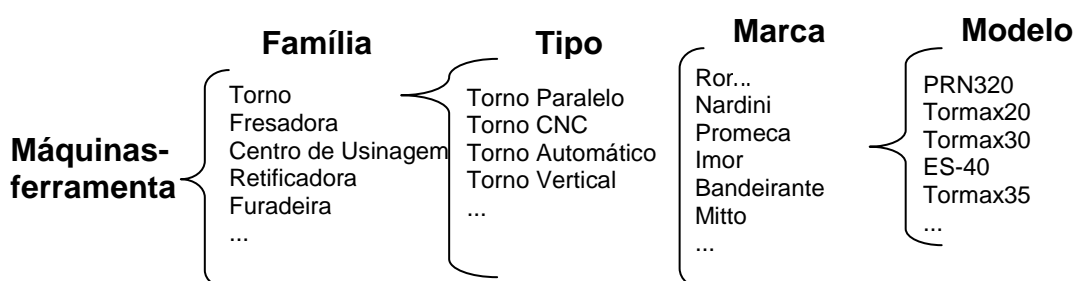


ILUSTRAÇÃO 14 - Modelo de representação orientada a objetos de máquinas-ferramenta

3.3.3 Regras

Dentro deste contexto existem as regras que sempre serão definidas para o conjunto *feature* e máquinas-ferramenta, são regras que não possuem variáveis e associam atributos de valores constantes. Pode-se chamar de regras fixas as regras que já associam certo tipo de máquina-ferramenta durante o momento da criação de uma *feature* (cadastro). Por exemplo, uma das regras fixas, que sempre existirão, liga a *feature* e sua cinemática a cinemática de um conjunto de máquinas. Isto permite diferenciar as máquinas além da nomenclatura formal.

Em uma busca simples de máquinas-ferramenta, tanto o especialista quanto o usuário podem utilizar seus conhecimentos para buscar máquinas-ferramenta através da nomenclatura comercial. Uma estrutura em árvore, como árvore de objetos da orientação a objetos, se ramifica conforme haja variações de cada função, família, tipo, marca e modelo. Ao atingir o modelo de máquina tange praticamente todas as máquinas do mercado utilizadas atualmente, salvo, somente, máquinas modificadas, que, são contempladas através da modificação da cinemática funcional destas máquinas, permitindo que esta seja capaz de efetuar uma *feature* diferenciada. Deste modo, o especialista classificaria a máquina normalmente, porém acrescentaria a cinemática específica da máquina-ferramenta modificada através da inserção da nova *feature* que esta máquina é capaz de executar.

Visivelmente, unir os campos de atributos de máquinas com *features* parece mais simples. Porém, considerando o aspecto de atualização dos dados e a incrível dinâmica desses na engenharia em relação aos seus atributos, como, por exemplo, a inserção de um indicador adimensional, pode tornar todo o trabalho atual pouco proveitoso. Pensando nisso, na dinâmica de planejamento de processos e na sua evolução, a proposta de regras foi separar as regras intrínsecas das regras que possuem esta dinâmica. Nomearam-se como regras fixas e regras variáveis, explicadas, a seguir.

3.3.3.1 Regras fixas

Regras fixas são as regras que são criadas para ligar a *feature* à máquina. Estas regras, uma vez criadas não podem ser editadas, pois só é possível destruí-las eliminando a *feature*. Do ponto de vista do especialista, ao criar uma *feature* ou

uma máquina-ferramenta, ele é capaz de editar sua cinemática. Indiretamente seria a maneira como aquela *feature* seria fabricada. Mas, isto só é possível ao criar a *feature*, pois passa a ser um atributo característico da *feature*, então só é possível destruir a *feature* ou derivar uma *feature* desta já criada.

Na inserção de *features*, o especialista cria um conjunto de regras fixas e mais tarde as regras que variam conforme aspectos tecnológicos do conjunto *feature* + máquina. As regras fixas para *feature* são descritas a seguir:

- Eixos: corte, avanço e aproximação;
- Dimensões do material bruto (excluindo peso máximo, estimado para o aço); e
- Tipo da disposição da *feature* (única - simples, matriz quadrada, matriz circular).

Esse seria o conjunto mais simples de regras que ligariam a *feature* à máquina-ferramenta (uma vez escolhida a máquina). A partir deste ponto o especialista recebe um formulário para inserir outras regras que dispõem primeiramente sua *feature* e todas suas características. Junto a este formulário encontra-se uma "calculadora lógica" que dispõe comparadores lógicos, matemáticos e contextuais (para texto simples), onde é possível apontar para característica da máquina-ferramenta e/ou sua família, tipo, dentro de uma classe específica.

Devido à flexibilidade da linguagem SQL separam-se estes operadores em grupos lógicos (e, ou), aritméticos (mais, menos, multiplicação, divisão, resto da divisão), faixa de valores (>, <, maior e igual, menor e igual), igualdade (=, ≠), texto (uso de coringas na comparação).

3.3.3.2 Regras Variáveis

Ao contrário, as regras variáveis apontam para valores que serão definidos durante a busca. No instante em que o especialista indica um atributo da *feature*, coloca no formato variável, indicando que aquele atributo vai ser determinado durante a busca. Usando como exemplo um furo simples e uma furadeira, onde seu diâmetro pode ser relacionando com o diâmetro máximo de ferramenta suportado, a regra poderia ser montada da seguinte forma:

“SE Furo.Diâmetro <= D ENTÃO Selecione
Furadeira.Diametro_maximo_ferramenta>=D”

Nesse caso, todas as furadeiras existentes, de diâmetro máximo da ferramenta (Furadeira.Diametro_maximo_ferramenta), seriam selecionadas se o diâmetro do furo fosse menor que "D". E o diâmetro do furo (Furo.Diâmetro) é um atributo (ou característica variável) da *feature* que faz referência a uma dimensão (com ou sem limites) dentro da *feature* "furo". Este exemplo demonstra que regras variáveis somente serão aplicadas na instância da *feature* criada pelo especialista.

Entre muitas maneiras de armazenar regras, a escolha de uma se destaca pela facilidade de disponibilidade de material, a SQL. Focado nesta linguagem de acesso a base de dados, as regras de cadeia direta se convertem em cadeia reversa. Ou seja, a condição de seleção de determinadas máquinas seguem após restringir sua seleção.

Outra grande flexibilidade desta linguagem é o acesso a operadores lógicos. As cláusulas são armazenadas na sua forma final, o que permite simplesmente, ao recuperar de uma base de dados e utilizá-la na forma nativa da base SQL. Generalizando para um conjunto de atributos de uma *feature* (Fa) e um conjunto de atributos de máquinas (Ma) comparáveis entre si através de um operador lógico (Opr), o sistema seleciona uma lista de máquinas (MI) não sendo nenhum atributo de máquina explicitamente uma constante, pode-se demonstrar as regras:

SE Fa Opr Ma ENTÃO Selecione MI

Esta lista de máquinas MI seria continuamente depurada até que a última regra fosse executada, obtendo assim pelo menos uma máquina-ferramenta. Na linguagem SQL a mesma regra demonstrada anteriormente pode ser representada da seguinte forma:

SELECT MI WHERE Ma Inv_opr Fa

Observa-se que o operador Opr foi invertido (Inv_opr) para manter o relacionamento entre máquinas e *features*. Isso acontece porque o especialista preenche a base de dados do ponto de vista da *feature*. Porém, o usuário procura através de regras as máquinas que são os objetos alvo da busca.

O especialista se confronta, durante o cadastro, com as *features* e suas características. Outros valores constantes foram previamente preenchidos na base de dados na forma de limites, máximo e mínimo, para um determinado atributo da *feature*. Ou seja, uma regra que possui uma constante é dividida em duas regras, a que associa a *feature* ao tipo de máquina-ferramenta diretamente e a que restringe algum atributo da *feature*:

SE *Feature*.DimensaoA > 1 ENTÃO Selecione Máquina.Tipo

Ou em SQL

SELECT Máquina.Tipo WHERE *FeatureID* = *Feature*

E

UPDATE *Feature* SET Dimensao_maxima = 1

Casos onde existam faixas preestabelecidas pode-se estimar campos limites para um atributo, podendo chamá-los de máximo para o limite superior e mínimo para limite inferior:

SE *Feature*.Atributo >= 1 ENTÃO Selecione Máquina.Tipo

Em SQL:

SELECT Máquina.Tipo WHERE *FeatureID* = *Feature*

E

UPDATE *Feature* SET Atributo_valor_maximo = 1

Condicionalmente, se o atributo da *feature* for comparado com uma constante, os limites (máximo ou mínimo) daquele atributo são editados e a declaração se torna somente a seleção do grupo de máquinas especificado.

Sendo uma regra de produção, os atributos das *features* não podem ser comparados entre si, nem com a caixa envoltória do material bruto, pois gerariam regras inválidas para o usuário final. No primeiro caso, a comparação de atributos da *feature*, geraria uma relação de razão geométrica, podendo limitar a utilização da máquina.

No segundo caso, a razão da *feature* com a caixa envoltória do material bruto, geraria uma razão de proporcionalidade, que já teria sido contemplada, se a caixa envoltória do material bruto coubesse na máquina e a máquina fosse capaz de executar a *feature*.

3.4 Fluxogramas de Atividades

Cada acontecimento do diagrama de casos da ILUSTRAÇÃO 8 se desmembra em um diagrama de atividades, aqui exemplificados por meio de fluxogramas.

3.4.1 Sistema de Cadastro

Continuamente os resultados são armazenados e filtrados até que a última seqüência seja executada.

O fluxograma apresentado na ILUSTRAÇÃO 15, a seguir, tenta exemplificar a criação de um conjunto *feature*-máquina. Lembrando que durante a criação de uma regra devem-se distinguir os três tipos de regras: a primeira, se o operador é reversível matematicamente; a segunda, se atributos da *feature* estão comparados entre si; e a terceira se é um atributo do material bruto.

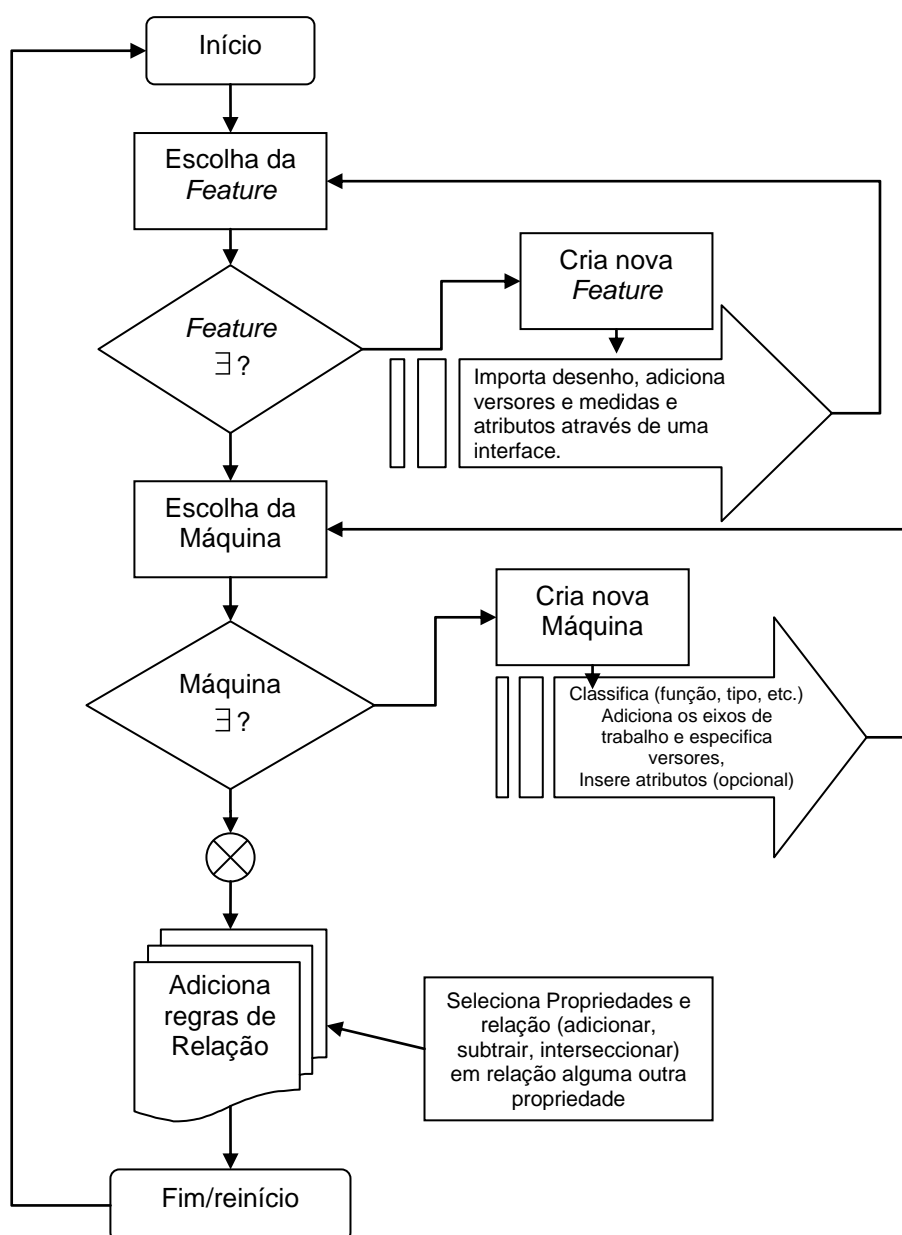


ILUSTRAÇÃO 15 - Fluxograma de criação das regras de produção

Durante a criação de uma *feature* nova, opcionalmente, é possível importar a representação gráfica desta, ou herdar de uma já existente. Conforme ILUSTRAÇÃO 16, deve-se adicionar ou modificar algum atributo com opção de remover outros. Seguindo a edição e posicionamento dos eixos que representam a cinemática desta.

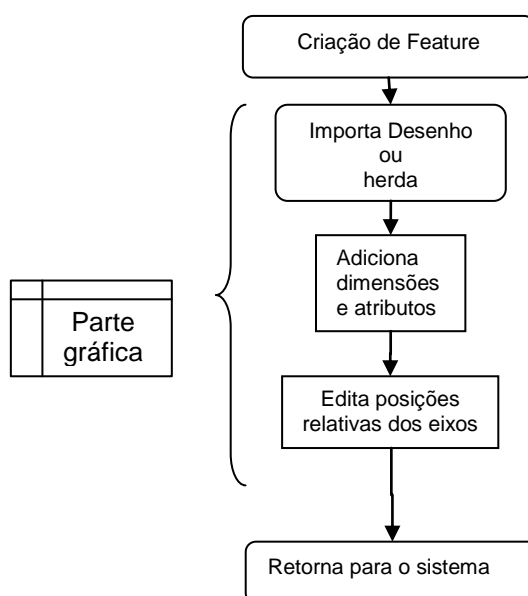


ILUSTRAÇÃO 16 - Adição de uma nova *feature*

Seguindo o fluxograma, a associação de uma máquina, é possível adicionar uma nova máquina (ILUSTRAÇÃO 17). Novamente com a opção de herdar propriedades de outra máquina durante a criação do perfil da máquina atual, adicionando, modificando ou excluindo atributos.

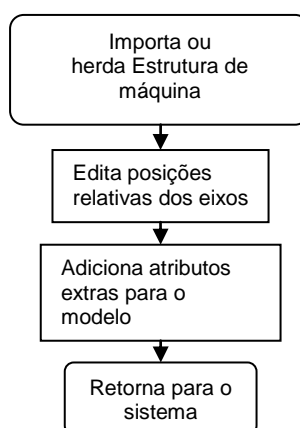


ILUSTRAÇÃO 17 - Adição de uma nova máquina-ferramenta

Finalmente, durante a criação da regra, onde se faz necessária a experiência de um especialista (ILUSTRAÇÃO 18). Internamente ao sistema, esta

regra é convertida em cadeia reversa para representação no formato da linguagem SQL.

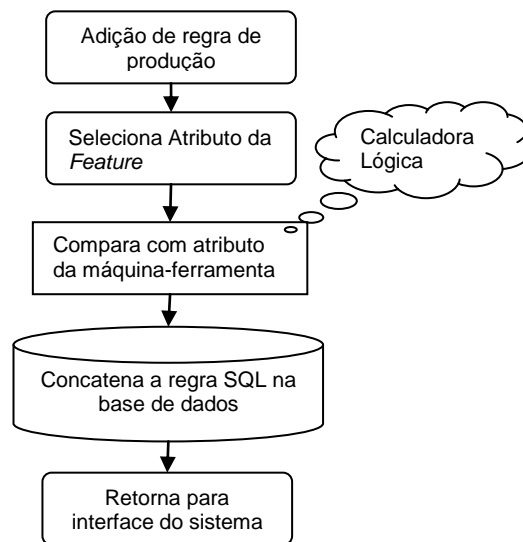


ILUSTRAÇÃO 18 - Adição de uma regra a *feature*

No instante que o processista especialista cria uma nova regra do ponto de vista da *feature*, esta regra deve ser convertida para o ponto de vista da busca da máquina-ferramenta. A conversão de cadeia direta em cadeia reversa, apresentada na ILUSTRAÇÃO 19, deve ocorrer em tempo para que a regra seja armazenada em um formato inteligível pela máquina de inferência.

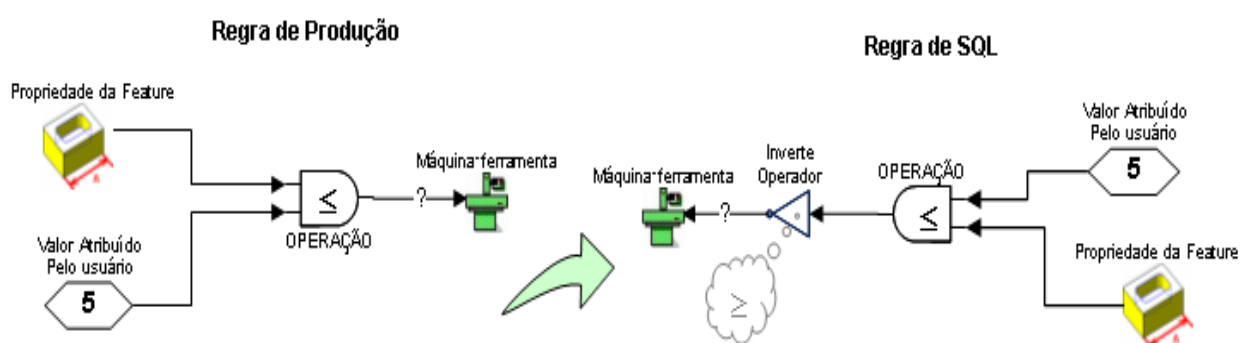


ILUSTRAÇÃO 19 - Conversão de cadeia direta em cadeia reversa.

Esta conversão deve ocorrer onde um filtro de operador é aplicado, para definir se realmente a regra possui algum tipo de comparação inválida, observadas detalhadamente na ILUSTRAÇÃO 20 e ILUSTRAÇÃO 21, a seguir.

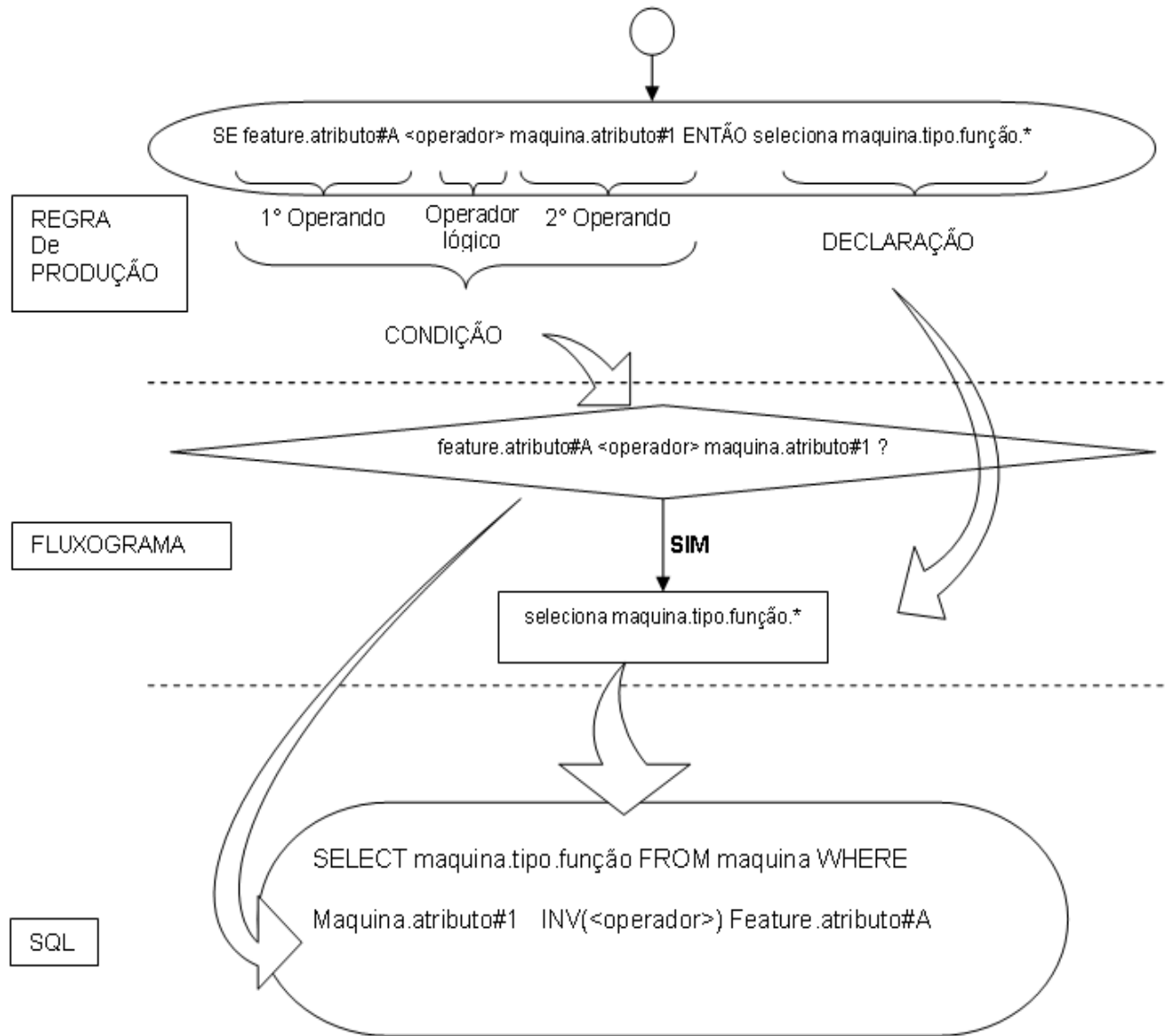


ILUSTRAÇÃO 20 - Detalhe da conversão das regras de cadeia direta em cadeia reversa (SQL)

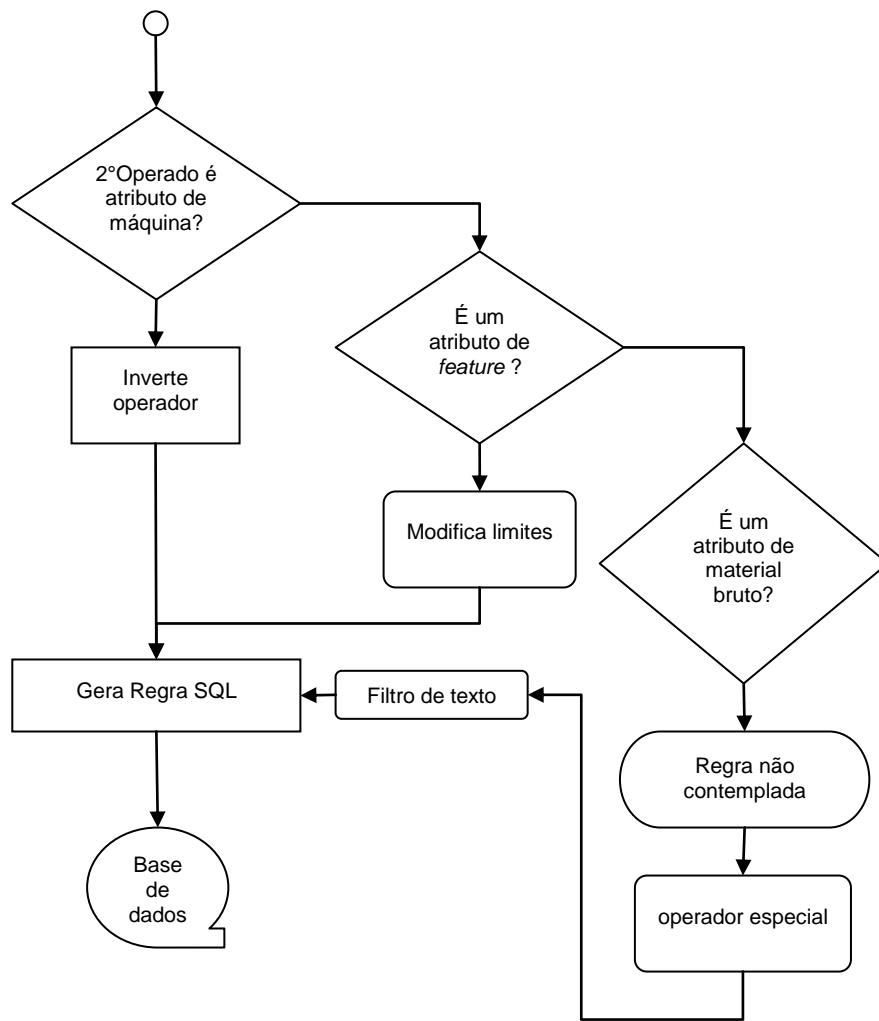


ILUSTRAÇÃO 21 - Algoritmo de separação das regras de produção

Concluída a preparação dos dados, a utilidade é fornecida pelo usuário do sistema que busca uma máquina-ferramenta através de regras de produção.

3.4.2 Sistema de busca

De fato a inferência só ocorrerá quando o usuário, em um determinado momento, instanciar, ou seja, definir a *feature*, e efetivar a busca. Nesse instante, acontece a recuperação da montagem das regras em seqüência na ILUSTRAÇÃO 22, preenchendo com valores faltantes e executando uma a uma até encontrar as máquinas, (ou a máquina), necessárias que são capazes de executar aquela operação conforme especificado.

Ao final, com a busca completamente definida, é possível, ainda, a inserção de parâmetros ligados diretamente à produção em relação às máquinas

encontradas. Com o intuito de refinar mais a busca, podendo citar: menor custo, máquinas ultrapassadas ou modernas, capacidade, tolerância de fabricação, disponibilidade da máquina, restrição de espaço geográfico, proprietário, entre outras.

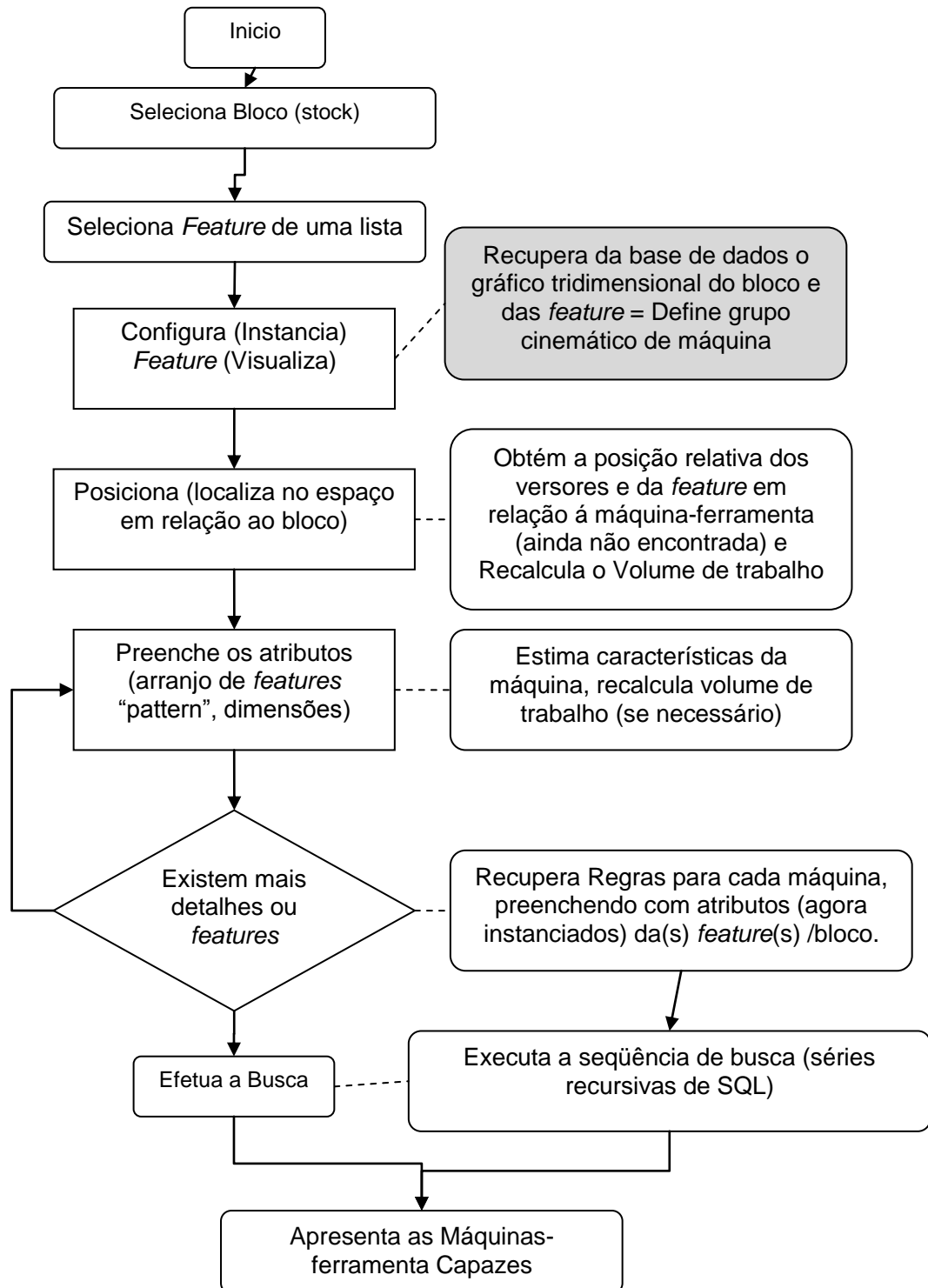


ILUSTRAÇÃO 22 - Fluxograma de busca de máquinas através de *features*

3.4.3 Inferência

Quando selecionar a caixa envoltória do material bruto e a *feature*, o usuário ainda não tem acesso a nenhuma máquina-ferramenta. Porém ao preencher algum parâmetro a seqüência de regras começa a ser indiretamente preenchida. O processo de preenchimento das regras e sua execução podem ser denominados inferência.

A REGRA 2 exemplifica este instante, que não é visível para o usuário leigo, onde ocorre a inferência de uma *feature* \mathfrak{F}_i , interno ao sistema, comparando os campos de eixos e modo de translação ("Trans_Modo").

```
SELECT MaqID FROM Maquinas WHERE FeatID.Ferramenta-ou-Peca = MaqID.Rotacao_na_Ferramenta-ou-peca AND
(
  SELECT FeatID.Eixos WHERE
    Vector.i = (
      SELECT Eixos.FeatID.Versor.i WHERE Eixos.FeatID = "{ $\mathfrak{F}_i$ }"
    )
  AND Vector.j = (
    SELECT Eixos.FeatID.Versor.j WHERE Eixos.FeatID = "{ $\mathfrak{F}_i$ }"
  )
  AND Vector.k = (
    SELECT Eixos.FeatID.Versor.k WHERE Eixos.FeatID = "{ $\mathfrak{F}_i$ }"
  )
) AND FeatID.TipoMov = MaqID.TipoMov AND FeatID.Trans_Modo = MaqID.Trans_Modo
```

REGRA 2 - Exemplo do processo de inferência no formato SQL como "subqueries"

Como resultado o usuário pode receber uma ou mais máquinas-ferramenta que atendem estas regras. Podendo eventualmente inserir mais restrições por meio de regras intrinsecamente associadas às *features*.

4 IMPLEMENTAÇÃO, RESULTADOS E DISCUSSÕES

Para exemplificar a metodologia em questão foi implementado um protótipo computacional simples, contendo as camadas de modelagem, manutenção e utilização apresentada na ILUSTRAÇÃO 9 e as interfaces para cadastros de regras máquinas e *features*, e busca. Entretanto o preenchimento das bases com o conhecimento referente às máquinas e *features* ficou limitado a construção de um caso simples para fins de teste do sistema proposto.

4.1 Validação

Para validação do sistema foram utilizados dados de exemplo, e um estudo de caso, descrito no QUADRO 3 e QUADRO 4 comentados a seguir:

Marca	Modelo	Função	Família	Tipo
NARDINI	UNK-01	FRESADORA	VERTICAL	DE CABEÇOTE MULTI-ORIENTÁVEL
Romi	prn-320	TORNO	HORIZONTAL	CONVENCIONAL
Romi	Discovery 4022	CENTRO DE USINAGEM	VERTICAL	CNC

QUADRO 3 - Dados de exemplo (máquinas utilizadas)

Marca	Modelo	Potencia Watts	Curso Horizontal	Curso Transversal	Curso Vertical	PesoMaxnaMesa
Romi	Discovery 4022	5670	320	250	350	750

QUADRO 4 - Dados utilizados, definição de uma das máquinas utilizadas

O projeto de base de dados foi feito com tecnologia de exportação, ou seja, a base de dados foi planejada em um software específico (Erwin v4.0) e posteriormente os scripts de criação foram gerados no servidor SQL (Microsoft Jet 4.0 e Microsoft ADO 2.7). O relacionamento dos dados poderia ser também repassado como regras de validação, que são disparadas a cada inserção de dados ("*triggers*"). A interface gráfica foi construída com auxílio do compilador VisualBasic (Microsoft) e módulos activex Cortona (Parallelgraphics).

Para geração das ilustrações tridimensionais, um software CAD com capacidade de exportar no formato VRML (ISO 14772 P1,2) adaptado manualmente com um editor de texto nativo do sistema operacional.

Para fins de teste foram selecionadas algumas *features* para fresamento e torneamento baseado na norma (ISO 14649 – P10, P12).

Quanto ao tratamento das caixas envoltórias, foram resumidas a dois tipos básicos: cilíndricas e prismáticas. Armazenados como referências dentro do programa (“StockID”) junto com suas medidas “*StockDims*”, que indiretamente limitam a máquina-ferramenta, pois possui medidas de contorno para comparar com a mesa da máquina.

O sistema permite inserir outras *features* e permite, também, cinemáticas diferentes de máquinas-ferramenta. Porém, nesta exemplificação foi resumida a cinemática da máquina como sendo CE cilíndricas associadas a máquinas-ferramenta de cinemática rotativa de peça (como torneamento) e CE prismáticas como sendo de cinemáticas de máquinas de mesa fixa (como fresamento).

As *features* todas foram baseadas na norma, e não nos volumes de remoção conforme Allada (2001). Para permitir o usuário visualizar sua posição real e dimensões de forma isolada criaram-se pequenas caixas prismáticas ao redor das *features* de fresamento e cilindros para as *features* de torneamento. Isto foi feito sem que impeça que o contrário seja feito (uso de *feature* de volume de remoção).

A inserção das cinemáticas de *feature* e máquinas-ferramenta estão separadas na interface gráfica, apresentada nas ILUSTRAÇÃO 23 até a ILUSTRAÇÃO 27, onde são apresentadas estas interfaces.

Um esquema de cores para especificar cada tipo de cinemática foi estabelecido, vermelho para sentido de corte, azul para o avanço e verde para o posicionamento. Infelizmente não foi possível comprovar neste trabalho a interferência entre o eixo de posicionamento e a peça. Isto garantiria a visibilidade da *feature* pela máquina-ferramenta, independente da ferramenta utilizada.

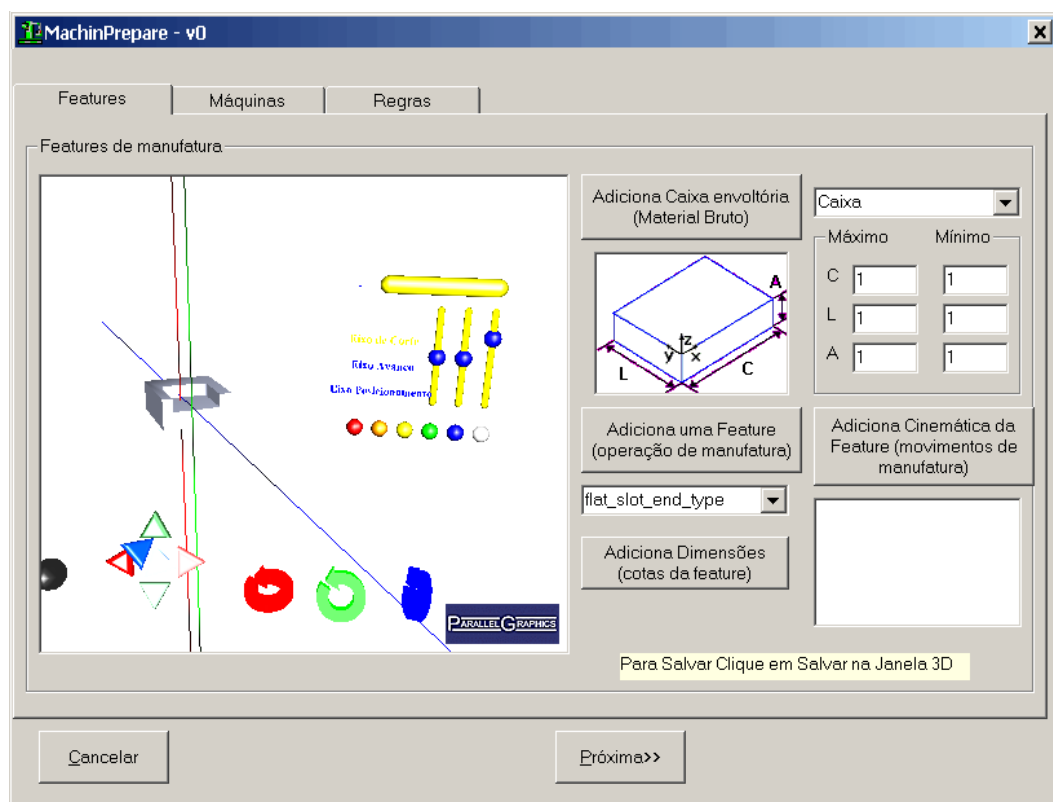


ILUSTRAÇÃO 23 - Interface de inserção da cinemática da *feature*.

Na ILUSTRAÇÃO 23 é possível observar também o quadro de inserção da caixa envoltória do material bruto, no canto superior à direita. De fato, isto dá início a especificação da *feature*, onde é criado material bruto que indiretamente limitaria a mesa ou fixação da máquina-ferramenta, assim como a capacidade de inserir cotas.

Estas cotas serão mais tarde repassadas como parâmetros para o sistema de inferência, na forma de regras, apresentado na ILUSTRAÇÃO 24, a seguir.

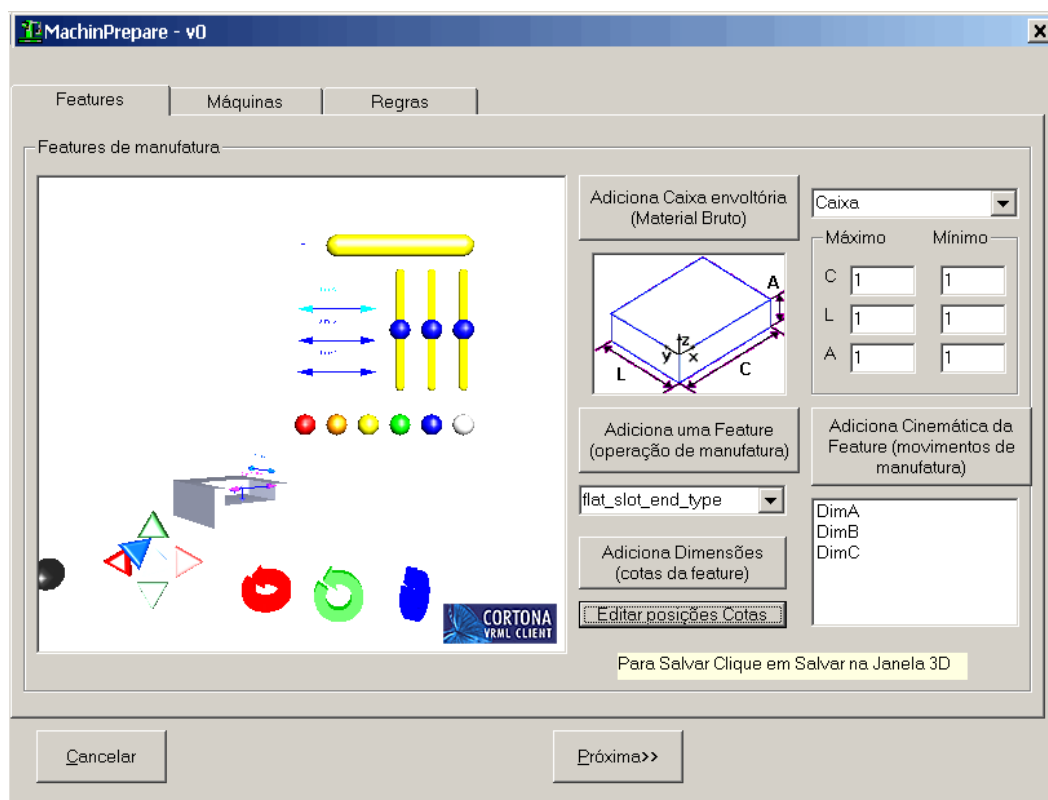


ILUSTRAÇÃO 24 - Inserção de cotas parametrizadas

As cotas são utilizadas para determinar regras ao usuário final que utiliza o sistema de busca. Estas podem trazer regras variáveis, como regra relacionando o limite máximo de trabalho da máquina e o comprimento máximo da *feature*.

Cada eixo da máquina ferramenta indiretamente é associado a *feature* correspondente através de regras de produção fixas. Essas regras só serão criadas caso o especialista, que utiliza o sistema de cadastro, selecione uma *feature* e uma máquina-ferramenta nas respectivas abas (do fichário ou *menu*) anteriores, podendo iniciar a edição de regras de produção.

Finalmente, na ILUSTRAÇÃO 25, a adição de máquinas-ferramentas, ligando a cinemática de cada eixo com a possibilidade de expansão das características desta máquina através da adição de atributos associados a valores.

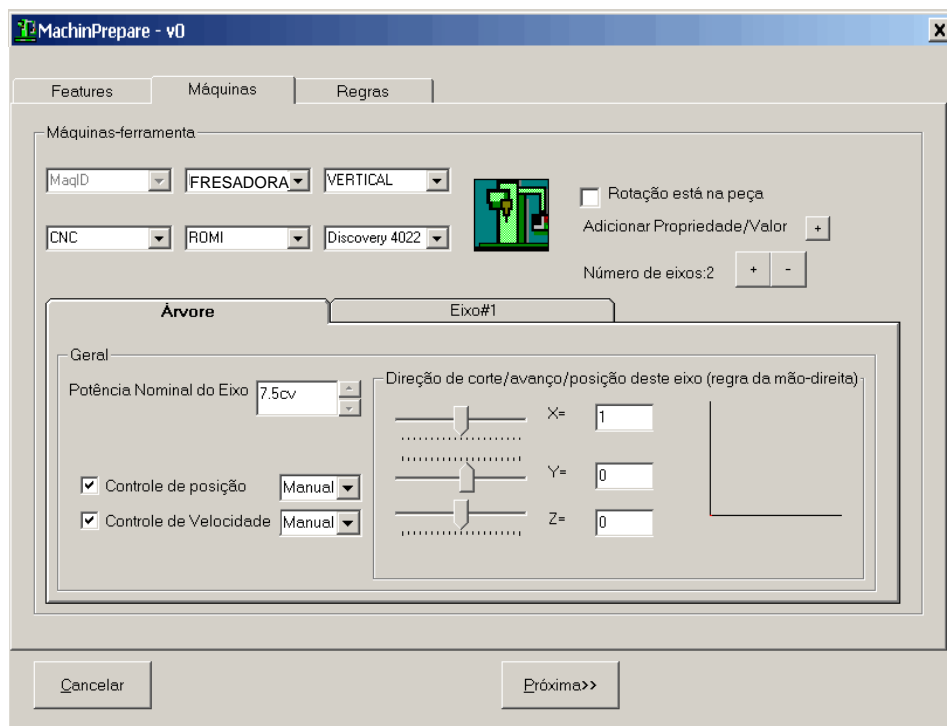


ILUSTRAÇÃO 25 - Edição de máquinas-ferramenta e cinemática

Partindo dessas premissas têm-se os seguintes resultados: um sistema de auxílio à busca de máquinas-ferramenta pode ajudar diretamente o especialista na tomada de decisão. Em grandes parques de máquinas, torna-se árduo o cadastro contínuo de várias máquinas e *features*, mas a eficiência do sistema após seu cadastro já é comprovada. Principalmente, o uso de regras de produção permite flexibilidade suficiente para que o especialista cada vez mais incremente a utilização das máquinas, permitindo tomar decisões mais precisas e não super-utilizar ou subutilizar algumas máquinas-ferramenta.

A inferência é gerada através da utilização de regras de produção. Apesar das dificuldades foi criada uma interface superficial para a manipulação de regras de produção, o módulo “Rules”, gerencia a alteração entra regras de produção e regras SQL.

As regras são inseridas a qualquer momento, até que o especialista tenha certeza que todas as regras estão corretas (ILUSTRAÇÃO 27). A opção de permitir ordenamento nas regras é sutilmente colocada na forma de ordem seqüencial (através de pequenas flechas ao lado da lista de regras - ILUSTRAÇÃO 26). Apesar de contemplada, esta relevância não foi avaliada.

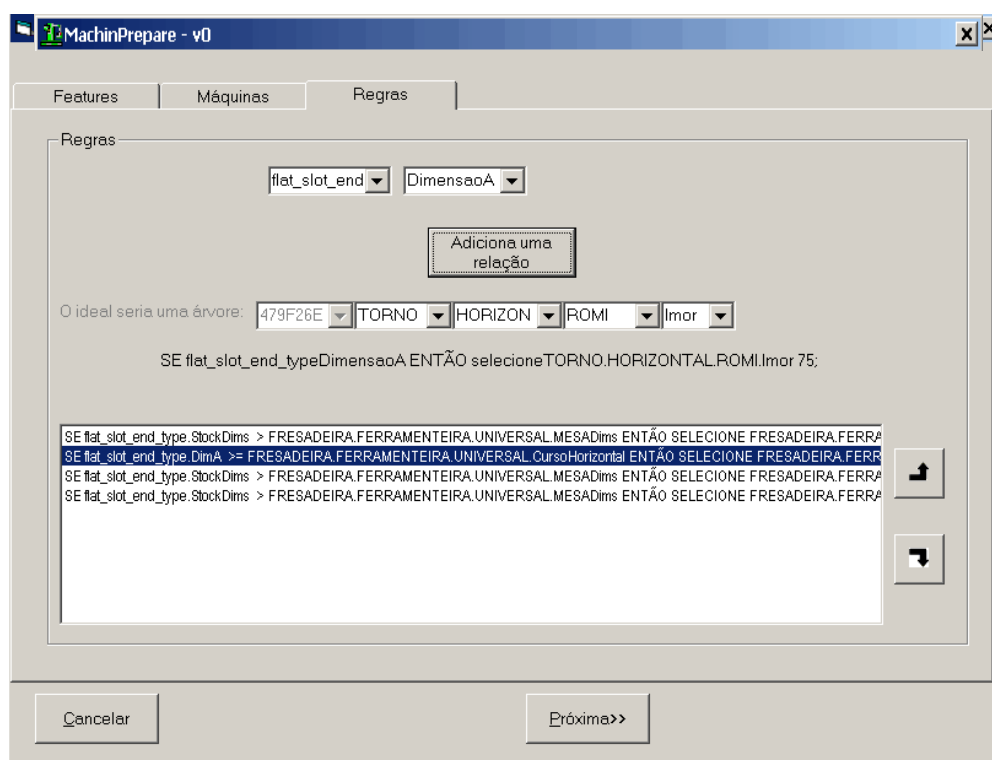


ILUSTRAÇÃO 26 - Visualização das regras de produção

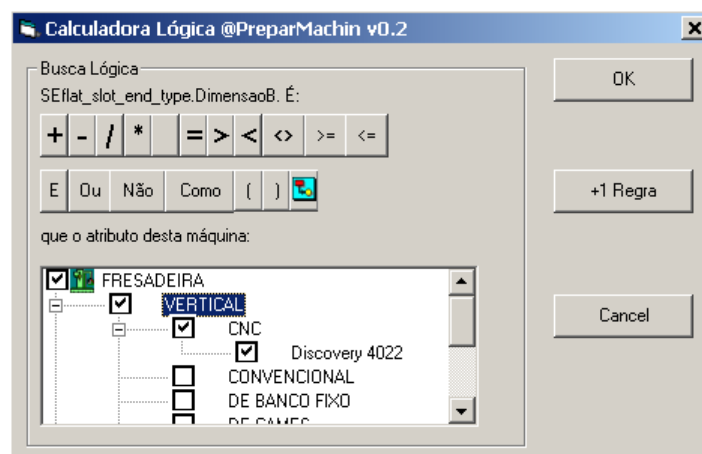
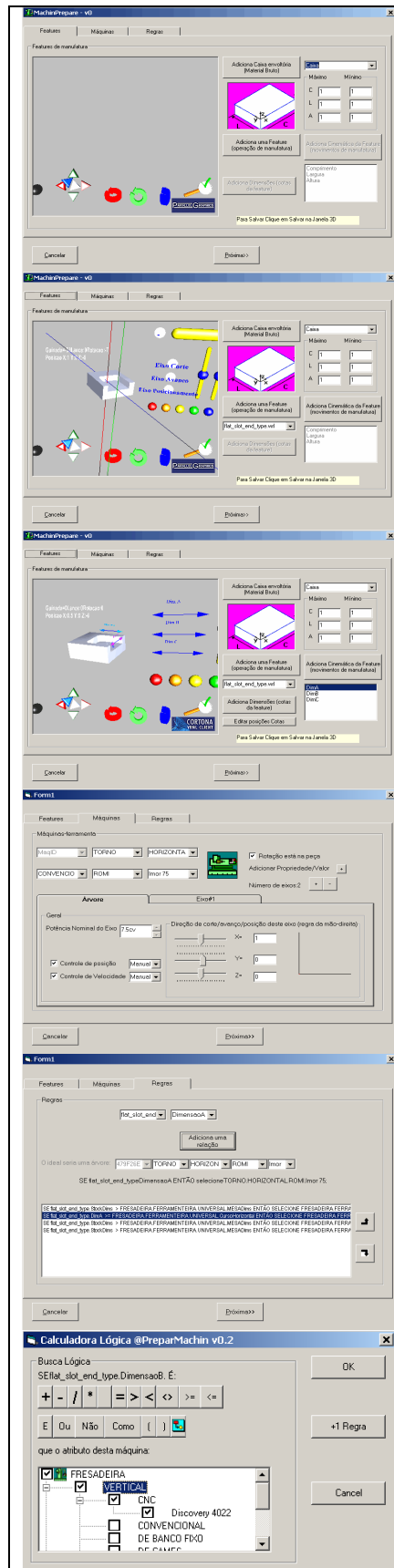


ILUSTRAÇÃO 27 - Cadastro de uma regra ("calculadora lógica")

No exemplo a seguir, apresentado na ILUSTRAÇÃO 29, demonstra-se a validação do sistema através de uma exemplificação. Somente as regras, por serem mais extensas, não serão mostradas.

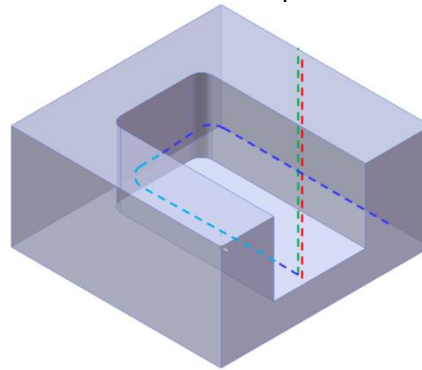


Seleciona a caixa envoltória

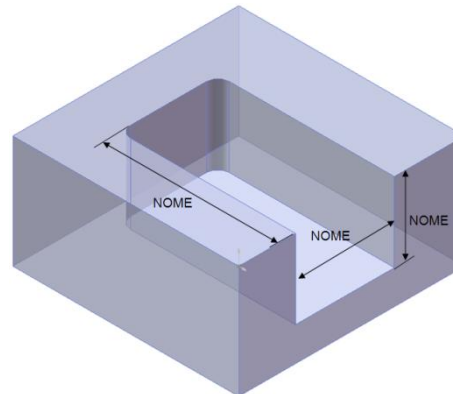


ou

Definem-se os eixos representativos



Posicionam-se as cotas:



Especifica a máquina (veja dados de exemplo)

Ao se adicionar regras, uma máquina específica ou um grupo de máquinas pode ser utilizado como parâmetro de busca.

ILUSTRAÇÃO 28 - Estudo de caso, criação de uma *feature* de fresamento

Pelo meio de uma contínua alimentação do sistema, a próxima etapa é a de utilização, onde se recuperam as regras na seguinte seqüência mostrada nas ilustrações a seguir:

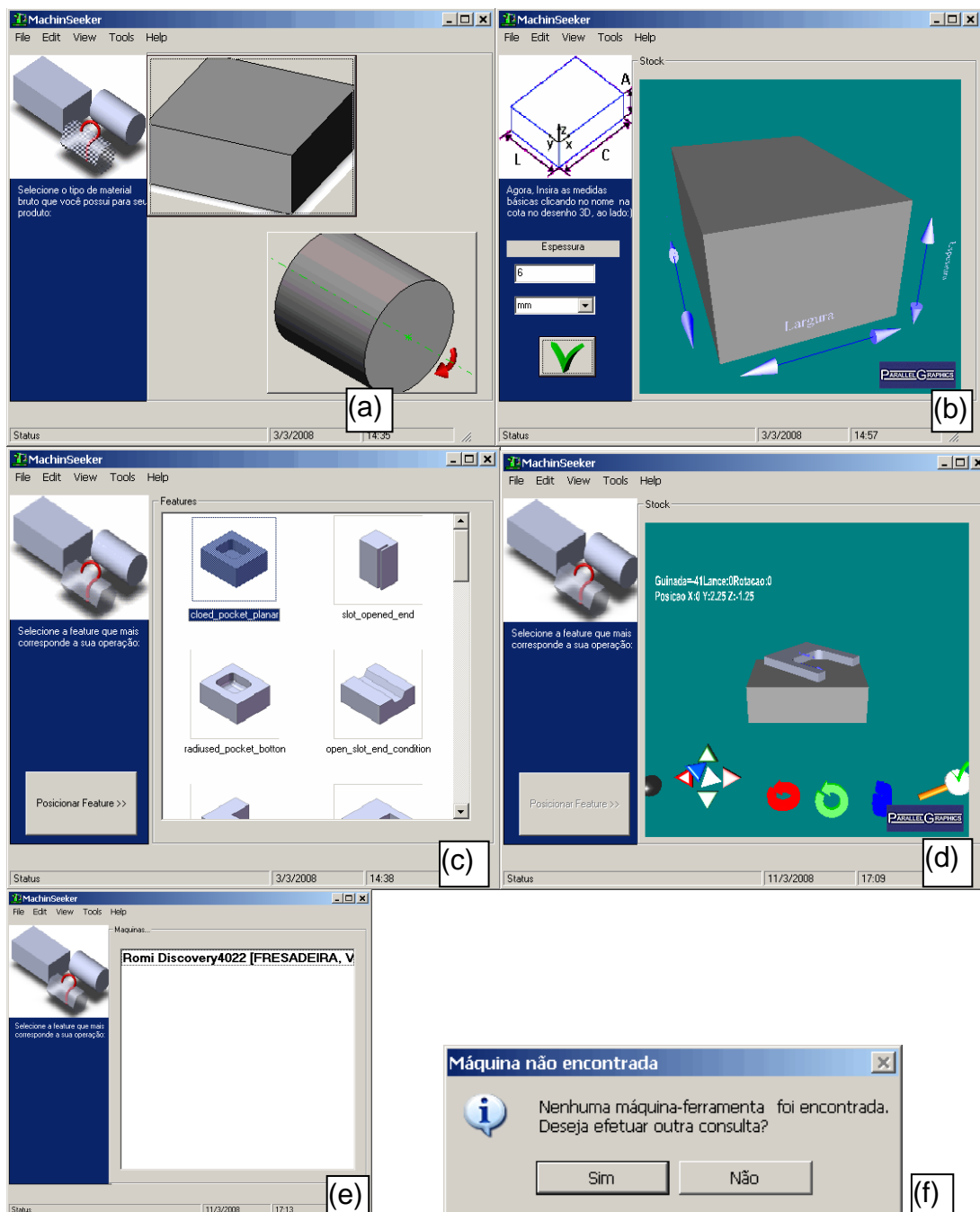


ILUSTRAÇÃO 29 - Telas do software na seqüência de seleção de *features* de fresamento.

A seqüência acima, representa a seleção do material bruto (ILUSTRAÇÃO 29a), logo é necessária a inserção das medidas (ILUSTRAÇÃO 29b) para limitar a capacidade da máquina. Uma série de ilustrações são apresentadas em seqüência, cada uma destas ilustrações representa (inicialmente) uma *feature* segundo a norma ISO 14649, seguindo as respectivas partes (P10 para o fresamento e P12 para torneamento) que definem o conjunto de possibilidades, apesar de limitadas, para o fresamento (ILUSTRAÇÃO 29c).

No início do posicionamento definem-se os eixos da máquina-ferramenta, a interface a seguir, que é tridimensional, permite armazenar a posição espacial da *feature* em relação ao material bruto (ILUSTRAÇÃO 29d).

Concluído e compreendido o posicionamento da *feature* em relação ao material bruto, os dados são armazenados e pode-se corrigir os eixos de posicionamento da máquina ou recalcular o reposicionamento do material bruto sobre a mesa/placa da máquina-ferramenta, este, não foi considerado neste protótipo.

Ao final (ILUSTRAÇÃO 29e) uma lista das possíveis máquinas-ferramenta aparecerá, a opção pessimista ocorre na ILUSTRAÇÃO 29f onde a opção de reinício da busca é apresentada.

Pode-se visualizar na ILUSTRAÇÃO 30 a estrutura final, no diagrama de visão lógica das classes e relacionamentos. No diagrama lógico, separam-se as classes que o usuário tem contato, das classes de trabalho de fundo e classes de dados.

As principais classes de trabalhos são a de *Features* (*features.mod*) e Máquinas (*maquinas.mod*), que gerenciam os dados durante e após a busca na classe de dados (*DBUsin.mod*). Os outros “pacotes” (*packages* do UML2) são conectores do tipo “ActiveX”, ou seja, ligam as classes aqui criadas a classes do sistema operacional, visualizam-se as principais “CortonaVRMLCliente” para visualização tridimensional e “ADODB” e “ADOX” de acesso a dados.

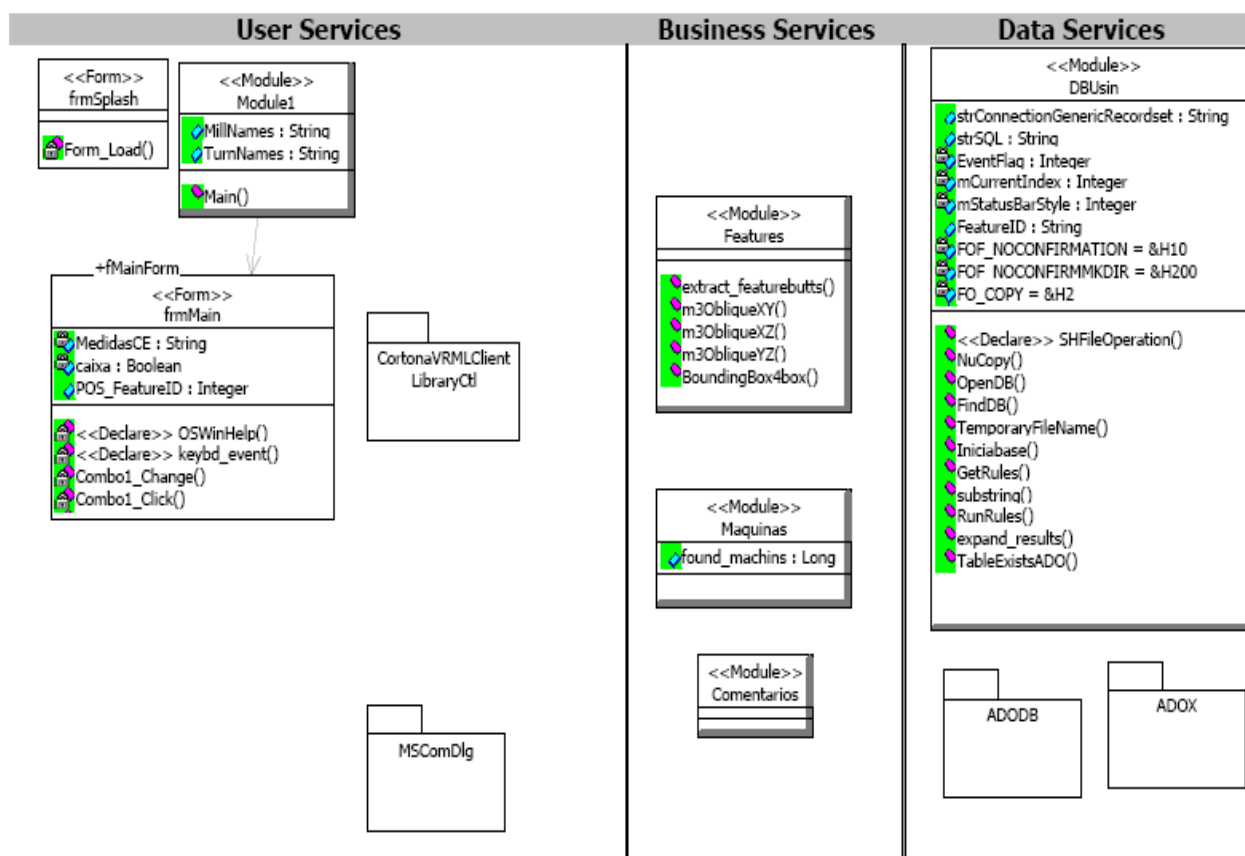


ILUSTRAÇÃO 30 - Diagrama de Classes do UML2, "logical view"

Apesar de ser constituída de uma base de dados robusta, esta foi subutilizada para permitir a exemplificação. A representação gráfica de versores rotacionais foi simplificada para permitir sua visualização, linhas substituem esta representação de versores rotacionais que parte de positivo no sentido horário para positivo no sentido da *feature*.

Outras restrições de busca (menor custo, maior produtividade) já citadas, não foram inseridas, pois o cadastro de máquinas-ferramenta depende que estes fatores tenham sido previamente definidos na base de dados para efetivar novas classificações, além da cinemática.

As principais limitações da utilização da linguagem VisualBasic® são a falta de portabilidade dos arquivos executáveis (.exe), restringindo a utilização das bibliotecas ActiveX® (.ocx) e a falta de ferramentas gráficas tridimensionais. Isto tornaria a versão atual deste software pouco aplicável em outros computadores e chão de fábrica. Como citado anteriormente na seção 2.5 outras propostas comerciais são apresentadas para armazenagem do conhecimento de usinagem.

5 CONCLUSÃO

O objetivo deste trabalho foi alcançado pela apresentação de uma metodologia para seleção de máquinas-ferramenta, baseada em características geométricas da peça e características geométricas e cinemáticas de máquinas-ferramenta. Um protótipo computacional baseado em uma simplificação desta metodologia foi implementado.

O uso de regras de produção embutido em cláusulas SQL se mostrou uma ferramenta flexível de programação e que permitiu a implementação e atualização de um sistema variante para a seleção de máquinas-ferramenta.

Embora o uso de regras de produção seja limitado, pois conduz a uma fragmentação do conhecimento representado, a utilização de cláusulas SQL contribui para o aumento da flexibilidade dessa técnica, possibilitando seu acréscimo e/ou atualização em qualquer estágio de utilização do projeto.

O emprego de *features* de usinagem baseadas na norma ISO 14649 foi útil como meio de junção da geometria e cinemática necessária a sua usinagem. A divisão em duas etapas de aplicação da metodologia (sistema de cadastro e sistema de busca) mostra que ainda é necessária a interferência humana tanto na criação de critérios de decisão como na tomada de decisão propriamente dita.

E, também, com a utilização de *features* de manufatura normalizadas pode contribuir para facilitar o trabalho de implementação de sistemas variantes.

A seleção de uma máquina-ferramenta é uma tarefa não trivial que demanda uma grande experiência do planejador. A construção de um sistema para planejamento de processos que contemple essa tarefa requer como ponto de partida a modelagem da experiência acumulada pelos processistas. Foi possível mostrar que a associação dos conceitos de *features* e regras de produção possibilitam a representação de parte desse conhecimento.

A metodologia apresentada permite visualizar a utilização das regras de produção aplicadas a um sistema de busca de máquinas-ferramenta baseado em *features* de manufatura. A utilização de regras de produção pode ser expandida para dar suporte à tomada de decisão.

É possível, ainda, aproveitar todo ou parte dos módulos implementados do software, devido a construção modular da metodologia.

5.1 Sugestões de Trabalhos Futuros

- Integrar bases de dados a sistemas existentes ERP; e
- Permitir que o sistema seja acessado por meio da internet (web).

6 REFERÊNCIAS BIBLIOGRÁFICAS

ABIMAQ, Associação Brasileira Da Indústria De Máquinas E Equipamentos. **Datamaq**. Disponível em: <<http://www.datamaq.com.br>>. Acessado em: 6 de fev. de 2008;

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Norma **NBR 6162**, Conceitos da técnica de usinagem - Movimentos e relações geométricas, Rio de Janeiro, 1989;

_____. **NBR 6163**, Conceitos da Técnica de Usinagem – Geometria da cunha de Corte, Rio de Janeiro, 1989;

ABITEBOUL, Serge; HULL, Rick; VIANU, Victor. **Foundations of Databases**. London: Addison-Wesley, [S.I.]. 1995.

ACHERKAN, N., FEDOTYONOK, YERMAKOV, V. **Machine Tool Design. Vol. 2**, 1973, Translate from Russian by Nicholas Weinstein, Honolulu: Ed University Press of Pacific Honolulu, Hawaii. 2000.

ADAMCZYK, Z. **A new approach to CAM systems development for small and medium enterprises**. Journal of Materials Processing Technology, vol.107, p.173-180, [s.n.] 2000.

ALLADA, Venkat. **Computer-Aided Design Engineering and Manufacturing - System Techniques and Analysis**. Vol.5, C.2, CRC Press, [S.I.]. 2001.

ALTING, Leo; ZHANG, Hongchao. **Computer Aided Process Planning: the state-of-art survey**. International Journal of Production and research, v.27, Elsevier, [S.I.], 1989, p553-585.

ARSLAN, M. Ç., ÇATAY, B., BUDAK, E. **A decision support system for machine tool selection**. journal of manufacturing technology management, Vol. 15, no. 1, 2004, p.101-109,.

ASHBY, M.F.; BRÉCHET, Y.J.M.; CEBON, D.; SALVO, L. **Selection strategies for materials and process**. Material & design, Vol. 25, 2004, p.51-67.

AVGOUSTINOV, Nikolay. **VRML as means of expressive 4D illustration in CAM education**. Future Generation Computer Systems, vol.17, London: Ed Elsevier. 2000. p39–48.

BABIC, Bojan; NESIC, Nenad; MILJKOVIC, Zoran. **A review of automated feature recognition with rule-based pattern recognition**. Computers in Industry, London: Ed Elsevier. Sep., 2007. doi:10.1016/j.compind.2007.09.001.

BLACK, J. T. **Traditional Machining Processes**. Metals Handbook, vol.16, 9th Ed., 2002.

BURBIDGE, John L. **Principles of Production Control**. London: Ed. Macdonald & Evans LTD, 3rd Ed. 1971. 485p.

CANCIGLIERI Jr., O. **Product Model Based Translation Mechanism To Support Multiple Viewpoints In The Design For Manufacture Of Injection Molded Products**. Doctoral Thesis, Loughborough University - Department of Manufacturing Engineering, Loughborough, United Kingdom. 1999. 369p.

CANG, Hsin-chi; LU, Wen F.; LIU, Frank X. **Machining process Planning of Prismatic Parts using Case-Based Reasoning and Past Process Knowledge**. Applied Artificial Intelligence, V.16, Ed.Taylor&Francis. 2002. p303-331.

CHANG, T., WYSK, R. A., WANG, H. **Computer Aided Manufacturing**. 2nd Ed., London: Prentice Hall. 2002.

CHIN, J., CHEN, Y. **System gains configuration and coordination of redundant degrees of freedom by genetic algorithms for multi-axis machine system in manufacturing**. Computers & Industrial Engineering, vol. 52, p355-379. Elsevier. 2007.

CHTOUROU, H., MASMOUDI, W., MAALEJ, A. **An expert system for manufacturing systems machine selection**. Expert Systems with Application, vol. 28. p.461-467. 2005.

CHUNG, C., PENG, Q. **The Selection of tools and machines on web-based manufacturing environments**, International Journal of Machinetools & Manufacture, Vol 44. p 317-326. 2004.

Da COSTA, D. D. **Proposta de uma Metodologia para Construção de um Modulo Especialista Dirigido a Escolhas das Condições de Corte**. Tese de Doutorado, Faculdade de Engenharia Mecânica, UNICAMP, Brasil, 1997.

Da COSTA, D. D.; NISHUMI, E. H.; MAGALHÃES, J. F. **Proposta Para Implementação De Uma Rede De Informações sobre Usinagem no Estado do Paraná**. Congresso Brasileiro de Engenharia de Fabricação (COBEF), Curitiba, 2001.

Da COSTA, S. C. **The prediction of risk of welding defects at the procedure stage using computer knowledge based systems**, PhD Thesis of Canfield Institute of Technology, Canfield, United Kingdo. 1992. 218 p.

DEGARMO, E. Paul. **Materials and processes in manufacturing**, New York, ed. McMillan, 8ht Ed. 1962.

DHANDE, S. G.; KARUNAKARAN, K. P.; MISRA, B. K. **Geometric modeling of manufacturing Process using Conjugate Geometry**, Journal of Engineering for Industry, Vol. 17, Agu. 1995. p. 288-296.

ESAWI, A. M. K. ; ASHBY, M. F. **The Development and Use of a Software Tool for Selecting Manufacturing Processes at the Early Stages of Design**. Proceedings of the Third Biennial World Conference on IDPT (Integrated Design

and Process Technology), Berlin, Germany, July 5 - 9, 1998, volume 3, pp 210 - 217.

FENG, C. Shaw. **Preliminary design and manufacturing planning integration using web-based intelligent agents**, Journal of Intelligent Manufacturing, vol.16, 423–437, Springer, Inc., Netherlands, 2005.

FLEISCHER, J.; DENKENA, B.; WINFOUGH, B., MORI, M. **Workpiece and Tool Handling in Metal Cutting Machines**. CIRP, Annals of, Feb. 2006.

GAO, J., X.; HUANG, X. X. **Product and Manufacturing Capability Modeling in an Integrated CAD/Process Planning Environment**. The international Journal of Advanced Manufacturing Technology, Vol. 11, p43-51, London: Ed. Springer-Verlag. 1996.

GHALLAB, Malik; NAU, Dana; TRAVERSO, Paolo. **Automated Planning - theory and practice**, ed. NY: Morgan Kaufman and Elsevier. US. 2004.

GRÉGOIRE, M. and SCHÖMER, E. **Interactive simulation of one-dimensional flexible parts**. Computer-Aided Design, Vol.39, Issue 8, Aug. 2007. p694-707

HAN, JungHyun; PRATT, Mike and REGLI, William C. **Manufacturing Features Recognition from Solid Models: A Status Report**. IEEE Transactions on Robotics and Automation, Vol. 16, No. 6, December. 2000. pg 782-796.

HOUTEN, Frederikus Jacobus Antonius Maria van. **PART: a computer aided process planning system**, PhD thesis, Univ. of Twente. 1991.

HSIEH, C. ; CHANG, T. **Motion fairing using genetic algorithms** Computer-Aided Design, Vol. 35, Issue 8, Jul. 2003. pgs739-749.

KNORST, Tarcísio. **Aplicação da tecnologia PFCP Integrada ao CAPP em ambiente de Ferramentaria**. Dissertação de Mestrado, UFSC, Florianópolis, Out. 1998. Disponível em: <<http://www.eps.ufsc.br/disserta98/knorst/cap4.html>>. Acessado em: 28 de nov. de 2007;

KUMAR, S.; SINGH, R. **A short note on an intelligent system for selection of materials for progressive die components**. Journal of Materials Processing Technology, vol.182, Elsevier. 2007. p.456–461.

KUNG, Kuna H.; DU, Timon C.; WENG, Cheng H. **Applying object-oriented database technologies in concurrent design process**. J. of Computer Integrated Manufacturing, Vol. 12, no.3. 1999. p. 251-264.

LEE, H. C., JHEE, W. C., PARK, H. **Generative CAPP through projective feature recognition**. Computer & Integrated Engineering, Vol. 53, Elsevier. 2007. p241-246.

LEPIKSON, H. A. Capítulo: **Sistemas Integrados de Manufatura, Tecnologias Avançadas de Manufatura**. 1 ed. Jaboticabal, SP: Novos Talentos, v. 1, 2005, p. 13-34.

LIN, Bor-Tsuen e HSUA, Shih-Hsin. **Automated design system for drawing dies**. Expert Systems with Applications, Vol.34, I.3, p.1586-1598. April 2008.

MARCIANO, João P. P.; BATALHA Gilmar F. **Notas de Aula de Planejamento de Processos de Fabricação - pmr5217**. Centro Politécnico da USP, São Paulo, Abr. 2004. Disponível em: <<http://www.poli.usp.br/d/pmr5217/>>. Acessado em: 28 de nov. de 2007;

NARAYAN, G. Aditya; PRAO Nalluri, S. R.; GURUMOORTHY, B. **Feature Based geometric reasoning for process planning**. SADHANA: Academy Proceeding in Engineering Sciences, Vol. 22. 1997. P2. p.217-240.

OBERG, E; JONES, F. D; HORTON, H.L.; RYFFEL, H. H. **Machinery's Handbook**. 27th Edition, London: Industrial Press. 2004.

PEKLENIK, J., SLUGA, A. **Contribution to Development of a Generative CAPP-System based on Manufacturing Process Topology**. Annals of CIRP, Vol. 38/1/1989, Jan. 1989.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Database management systems**. 2nd Ed. 2002.

RAPHAEL, B., SMITH, I.F.C. **Fundamentals of Computer Engineering**. West Sussex: Ed. John Wiley & Sons Ltda. England. 2003, 306p.

ROSSI, M. **Máquinas-operatrizes modernas**. Vol. I e II , Milão: Ed. Urlico Hoepli. 1970. 1061p.

ROSSO Jr, R. S. U.; NEWMAN, S. T. **Estrutura de Dados para sistemas CAD/CAM aderente a STEP**. VI CIBIM, Proceedings, Portugal, Coimbra. 2003.

SHAH, J. J. and MÄNTYLÄ, M. **Parametric and Feature-Based CAD/CAM**. New York: ed John Wiley & Sons. 1995.

SHAH, J. J. **Assessment of features next term technology**. Computer-Aided Design Vol.23, Issue 5, Jun. 1991. p331-343.

SILBERSCHATZ, A.; KORTH, H. F., SUDARSHAN, S. **Database Systems Concepts**. London: Ed McGraw-Hill. 2nd Ed. 1986.

SRINIVASAN, M.; SHENG, P. **Feature based process planning in environmentally conscious machining - Part 2: macroplanning**. Robotics and Computer Integrated Manufacturing, vol. 15, 1999. p.271-281.

SUDARSAN, R.; FENVES, S. J.; SRIRAM, R. D.; WANG, F. **A product information modeling framework for product lifecycle management**. Computer-Aided Design, vol.37. 2005. p.1399–1411.

TETI, R.; KUMARA, S. R. T. **Intelligent Computing Methods for Manufacturing Systems**. Annals of the CIRP, Vol. 46. Fev. 1997.

TIAN, Gui Yun; YIN, G.; TAYLOR, D. **Internet-based manufacturing: A review and a new infrastructure for distributed intelligent manufacturing**. Journal of Intelligent Manufacturing, vol.13. 2002. p323-338.

TSAL, Ping-ling; FU, Chi-Chang; CHIENG, Wei-Hua. **Analysis and synthesis of geometric models using tree-structured relations**. Computer-Aided design, Vol 29, No.9, GB. 1997. p. 607-615.

ZEID, Ibraim. **Mastering CAD/CAM**. Boston: Ed. McGraw-Hill, 3a. ed. US. 2005.

ZHOU, X.; QIU, Y.; HUA, G.; WANG, H.; RUAN, X. **A feasible approach to the integration of CAD and CAPP**. Computer-Aided Design, vol.39, Elsevier. 2007. p.324–338.

ZIMMERS, Emory W. **CAD/CAM Application in Machining**. Metals Handbook, ASTM, vol 16. 9th Ed. 2002.

INTERNATIONAL STANDARD ORGANIZATION. **ISO/IEC 14649**, P1,10,101, Data model for Computerized Numerical Controllers. Geneva. 2002;

_____. **ISO 10303**, P1,P11,AP203,AP238, Product Data Representation and Exchange. Geneva. 2000;

_____. **ISO 14772**, P1,2, The Virtual Reality Modeling Language. Geneva. 2004;

_____. **ISO 19775**, P1,2,3 Extensible 3D (X3D). Geneva. 2004;

_____. **ISO 19776**, P1,2,3, X3D encodings: XML encoding. Geneva. 2004;

_____. **ISO 19777**, P1,2, X3D language bindings. Geneva. 2004;